Support vector Machine Assignment

H00H3a

Master of Bioinformatics

Student name: Mianyong Ding

Student number: r0823572

Exercise Session 1: Classification

1.1 A simple example: two Gaussians

The randn function can return the normally distributed random numbers. The distribution of each class is two-dimension gaussian distribution, the centres of each class are (1,1) and (-1, -1) and they have the same co-variances. The optimal line could be y=-x+4, shown as figure 1. It can minimize the misclassification.

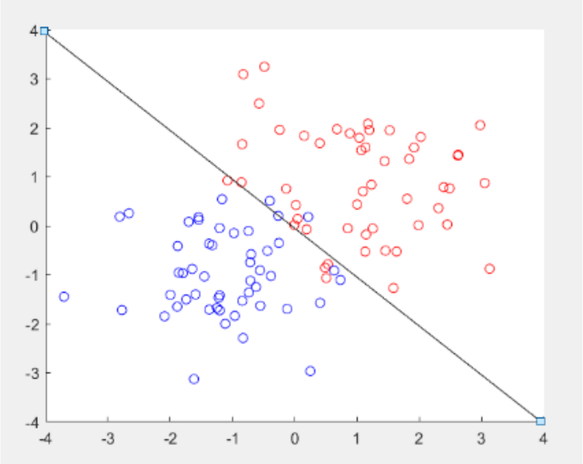
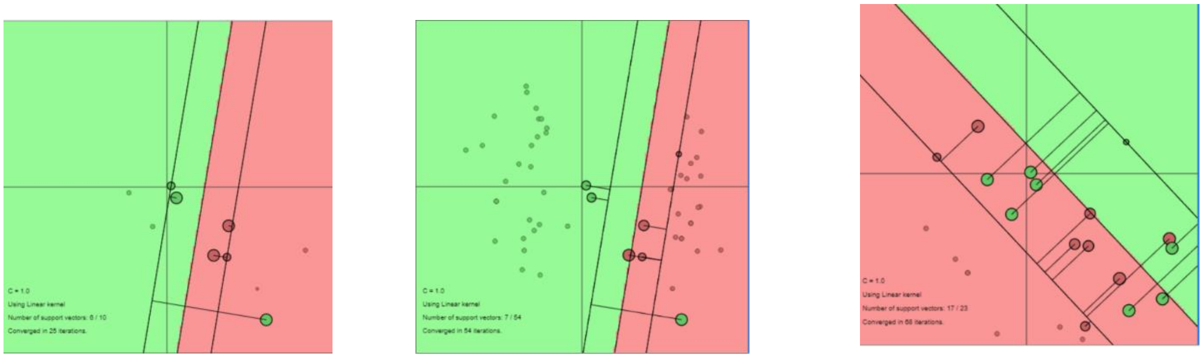


Figure 1. The two gaussians classification, colour represents different class.

2 Support vector machine classifier

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. If the data points who are close to the decision boundary, it is most likely to become support vector. The larger size represents higher importance to decide the position of hyperplane.



C = 1.0
Using Linear kernel
Number of support vectors: 6 / 10
Converged in 25 iterations.

C = 1.0
Using Linear kernel
Number of support vectors: 7 / 54
Converged in 54 iterations.

C = 1.0
Using Linear kernel
Number of support vectors: 17 / 23
Converged in 68 iterations.

Figure 2. SVM classifier with linear kernel. From left to right: the original data point. Middle: Add more data points on the right sides. Right: Add more data points on the wrong side.

If we add more data points on both right sides, we can see the scope of hyperplane is changed, but the margins and number of support vectors do not change. However, if we add the datapoints on the wrong sides, the margin, scope and support vectors have a dramatic changes.

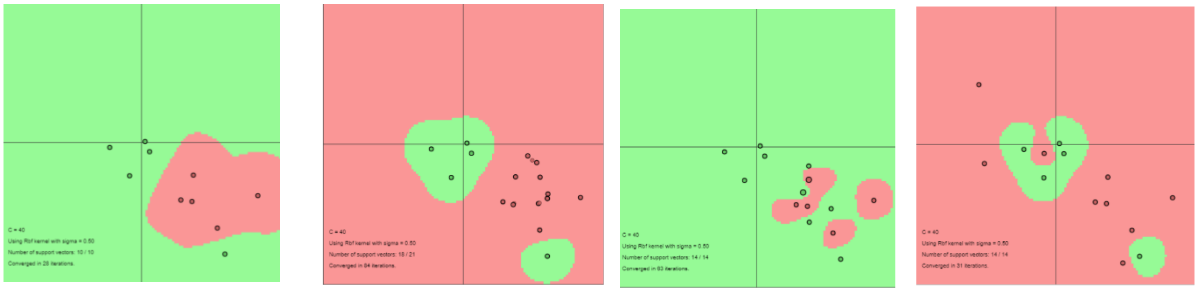


Figure 3. The SVM classification with RBF kernel. From left to right: The original data points. The addition of more red points on the right side. The addition of green points on the right side. The addition of red point on the wrong side.

For the RBF kernel, adding more data points on the right side can expand the region of class which the adding points belong. And the additions of data points of one class can lead to the “isolated island” of some data points belong to the other class. If we add more points on the wrong sides, we can observe the original regions are split to smaller isolated region.

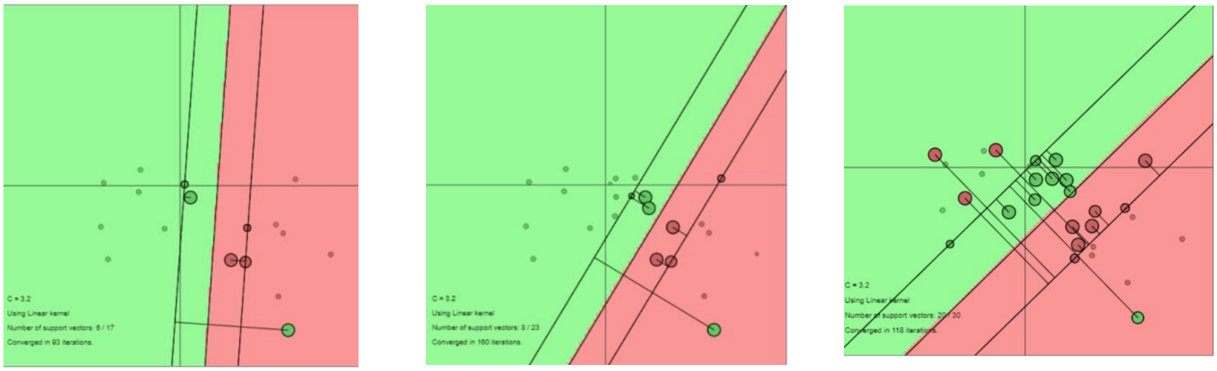


Figure 4. The SVM classification with linear kernel. From left to right: The original data points. The addition of more points on the right side. The addition of red points on the wrong side.

When there is a new point closer to the hyperplane, the importance of original support vectors will change. And, when there is some new addition of data points on the wrong sides, the importance of original support vector will also change. The C parameter can also influence the importance of support vector, as low C value will emphasis on the misclassified points.

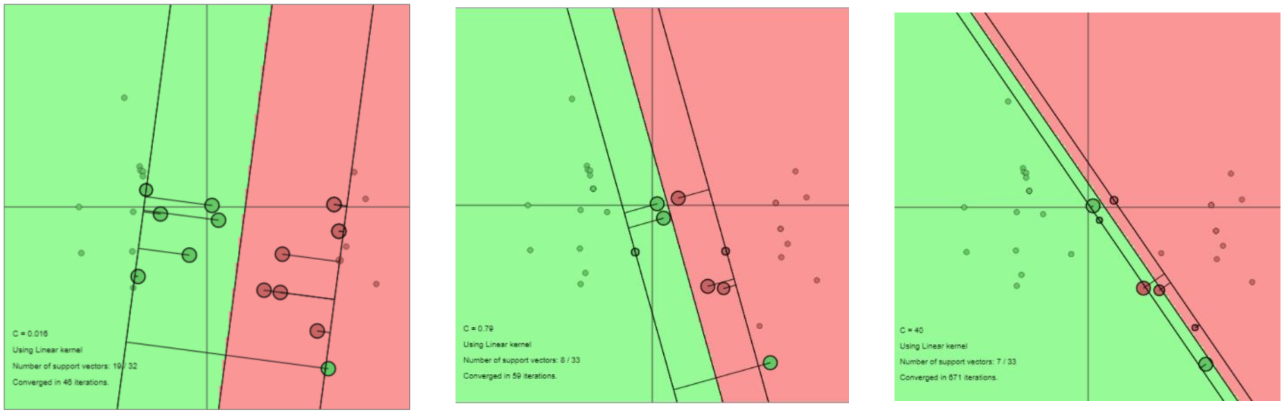
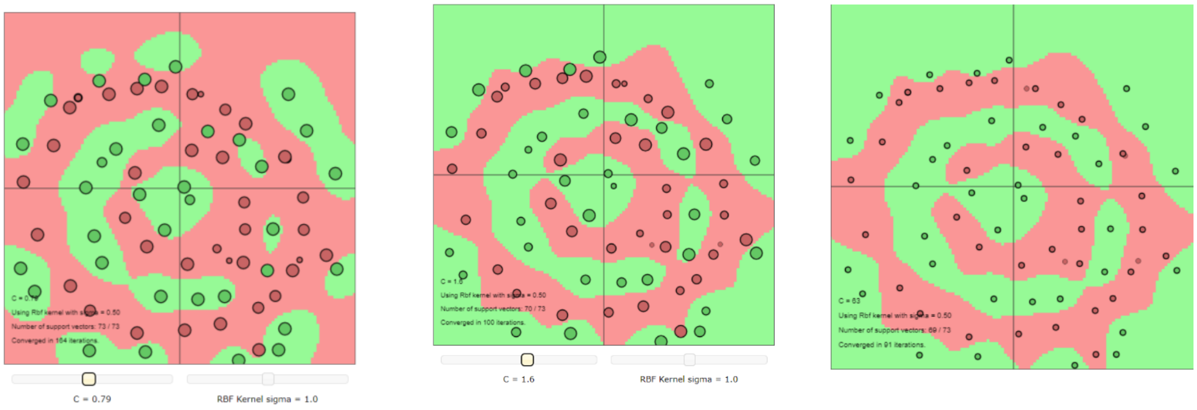


Figure 5. The SVM classifier with linear kernel for different choices of C parameter. (Left to right :0.015,0.79, 40).

From figure 5, we can observe that the margin expands with the decrease of value of C parameter, more support vectors are accounted. C parameter is the regularization parameter which controls the cost of misclassification on the training data. When C is larger, lower error is allowed. The C trades off the misclassification cost against the decision function's margin. If the C is large, making the high cost of misclassification (allow little error), leading to smaller the margin.

From our figure, with low C value, the green point which is in the red region is allowed. However, when the C value is large, the cost of misclassification is very high, the misclassified green point is not allowed. The model is strong influenced by this point.



C = 0.79
Using Rbf kernel with sigma = 0.50
Number of support vectors: 73 / 73
Converged in 184 iterations.
C = 0.79
RBF Kernel sigma = 1.0

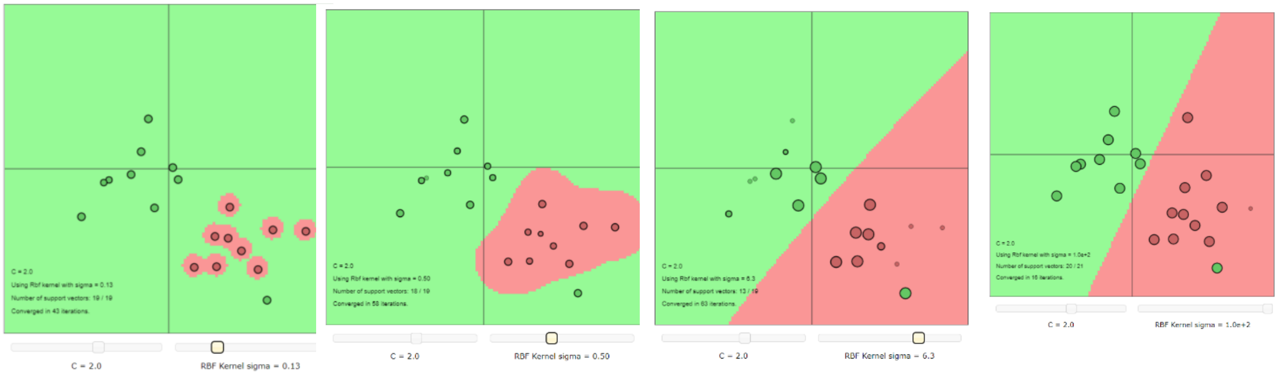
C = 1.6
Using Rbf kernel with sigma = 0.50
Number of support vectors: 70 / 73
Converged in 100 iterations.
C = 1.6
RBF Kernel sigma = 1.0

C = 83
Using Rbf kernel with sigma = 0.50
Number of support vectors: 69 / 73
Converged in 91 iterations.

Figure 6. The SVM classifier with RBF kernel for different choices of C parameter. (Left to right : 0.79,1.6,83).

For the RBF kernel, we can observe when the C is low, making emphasis on the misclassification errors, when the C is large, the decision boundary is smooth, the importance of support vectors becomes equal.

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



C = 2.0
Using Rbf kernel with sigma = 0.13
Number of support vectors: 19 / 19
Converged in 43 iterations.
C = 2.0
RBF Kernel sigma = 0.13

C = 2.0
Using Rbf kernel with sigma = 0.50
Number of support vectors: 18 / 19
Converged in 58 iterations.
C = 2.0
RBF Kernel sigma = 0.50

C = 2.0
Using Rbf kernel with sigma = 6.3
Number of support vectors: 13 / 19
Converged in 63 iterations.
C = 2.0
RBF Kernel sigma = 6.3

C = 2.0
Using Rbf kernel with sigma = 1.0e+2
Number of support vectors: 20 / 21
Converged in 16 iterations.
C = 2.0
RBF Kernel sigma = 1.0e+2

Figure 7. The RBF kernel with different sig2 value (0.13,0.5,6.3,100)

From figure 7, we can see when the sig2 is small, the decision boundary is non-linear. And the regions are isolated. Then the sig2 increase, the decision boundary starts to be linear. Sigma is to control non-linearity of model, the decision boundary is highly non-linear with the small value of sig2 and the decision boundary tends to be linear with the large value of sig2. Sig2 determine the reach of single training example; large values represent far reach of data points. With the low value, the decision boundary is determined by the closet data points. In contrast, large value will consider the datapoints which are far away. It introduces how much non-linearity is in our model. When sigma is very large, it cannot capture the shape of data, becoming linear decision boundary.

1.3 Least-squares support vector machine classifier

1.3.1 Influence of hyperparameters and kernel parameters

To test on the test sets, the misclassification rate is calculated. With the increase of degree, the misclassification error reduces. When the degree is 0, the error rate starts to be zero. However, the large degree may have the issues of overfitting.

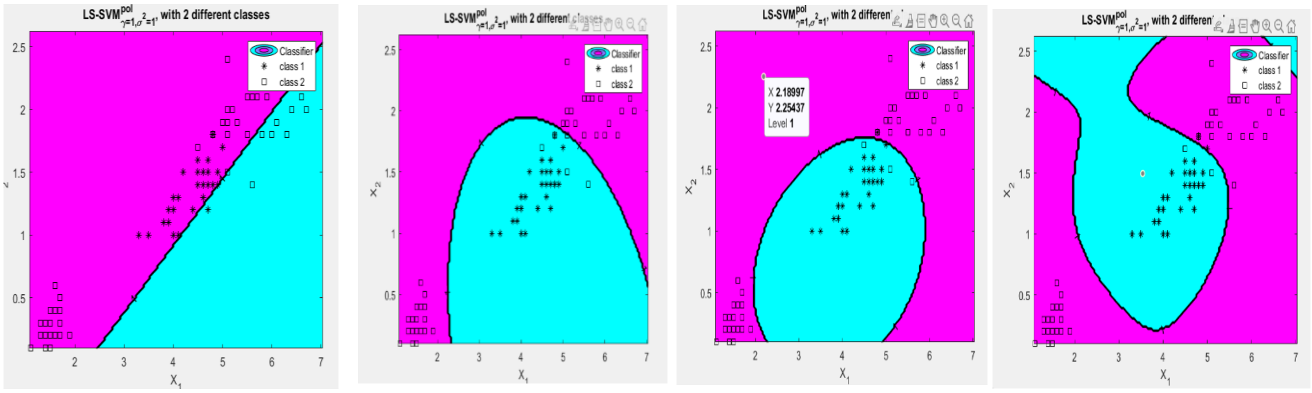


Figure 9. The polynomial kernel with different degrees (Left to right: Degree =1, error =55% ,Degree =2, error rate =5% ; Right: Degree =3, error rate =0 , Error rate =4, degree =0)

With the fixed Gam =1 the sig2 0.01,0.1,1 is tried, we can observe when the sig2 increase, the decision boundary becomes much smoother, and the classification error reduces to zero.

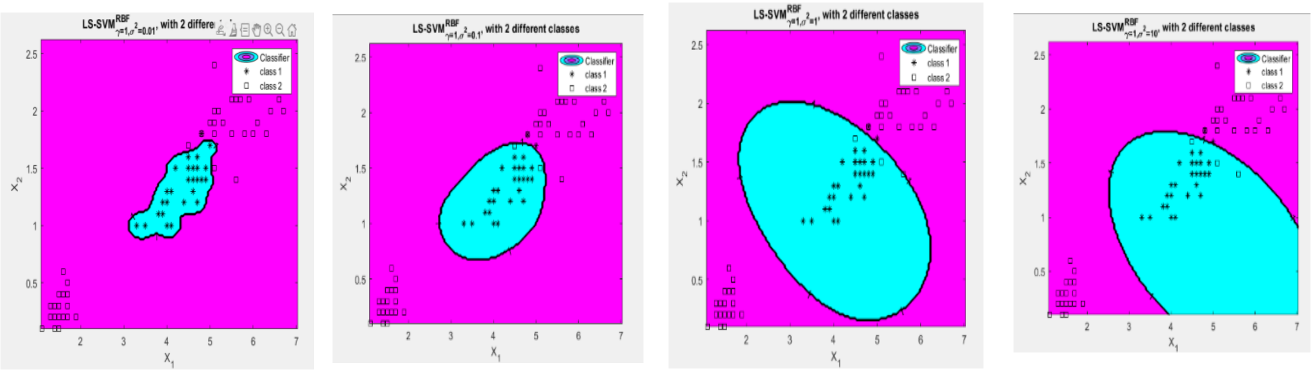


Figure 10. The different choices of sig2 with fixed gam (gam=1).

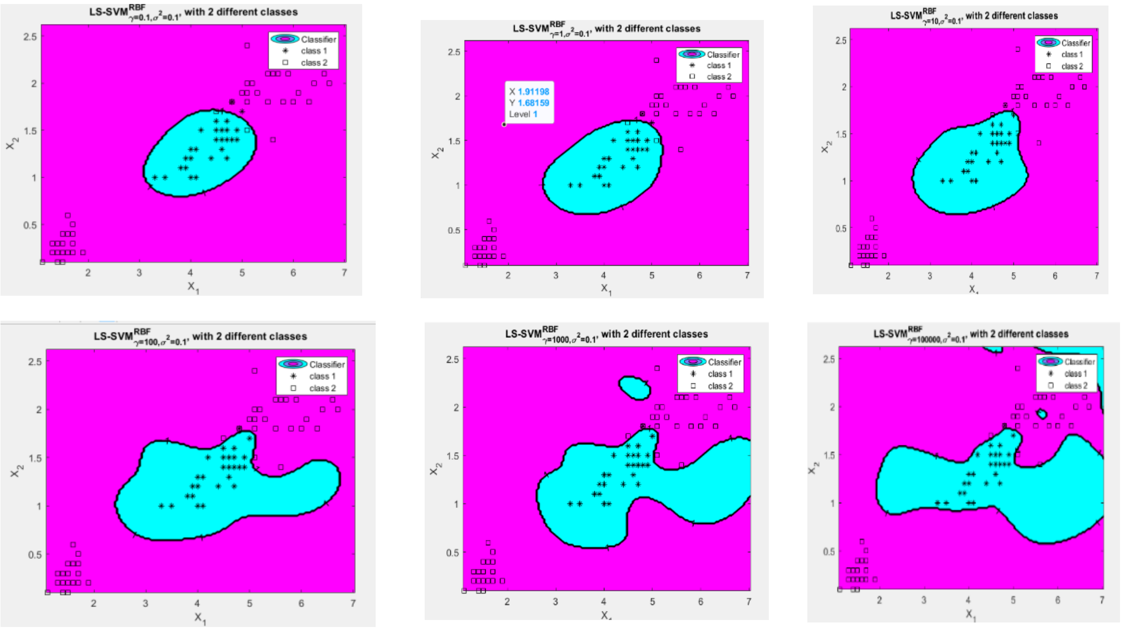


Figure 11. The different gam with fixed sig2.

gam	1	1	1	1	0.1	1	10	100	1000	10000	100000
Sig2	0.01	0.1	1	10	0.1	0.1	0.1	0.1	0.1	0.1	0.1
error	0.1	0	0	0	0.1	0	0	0.05	0.05	0	0

Table 1. The misclassification error of each combination of gam and sig2.

Gamma is the regularization parameter, determining the trade-off between the fitting error minimization and smoothness of the estimated function. We can see when the gam increases, the classification error tends to decrease globally. But when gam=100 and gam=1000, there is one point misclassified. The model is overfitted when gam is very large. The ideal ranges from the gam is around 1-10, and the ideal range of sig2 is around 0.1-10.

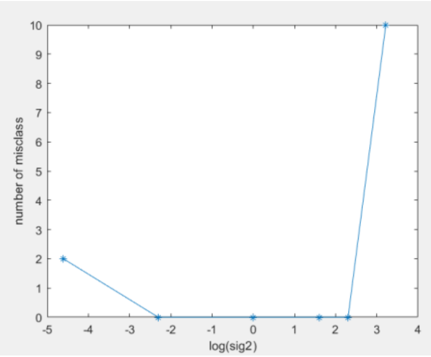


Figure 12. The misclassification of different value of sig2 from samplescript_iris.m.

The figure is from sample script, with the fixed gam, we can observe the ideal sigs is ranged for sig2 is 0.01 to 10 where the misclassification error is zero.

1.3.2 Tuning parameters using validation

In order to select the best range of gam and sig2, I select 60 gams and 60 sig2s to test the performance of total 3600 combinations. The values of gams and sig2s are expressed as e^(power). The selected value of power is [-30:1:30]. The heatmaps are generated for each validation methods, deeper color represents high error.

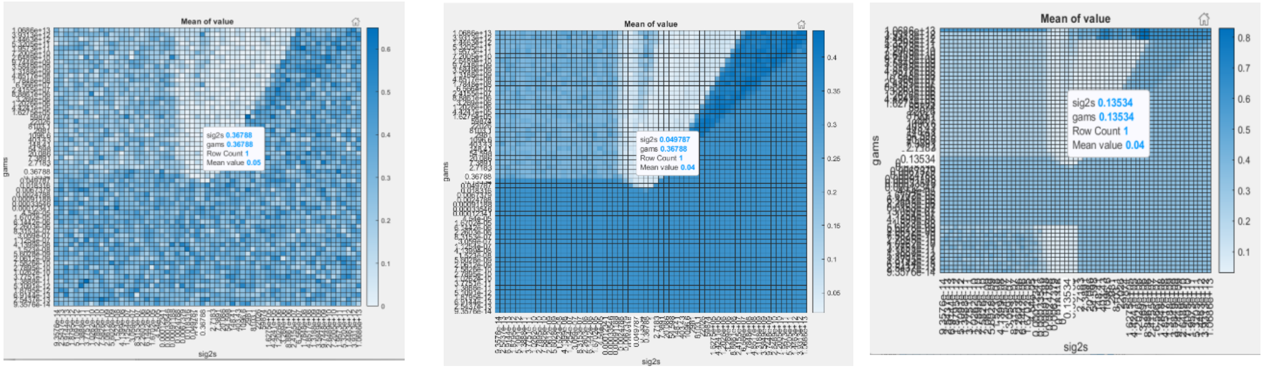


Figure 13. The performance of each validation methods (left to right: Random split, k-fold, leave one out). The color represents the error.

Comparing the results from three validation methods, we can observe the changes of color of k-fold method and leave one out method is more continuous. For the random split method, the color of many cells is different from their neighbor cells. The gam and sig2 from the region with smallest error (white color) are chosen as the optimized parameter. For random split, the gam around 0.36 and the sig2 around 0.36 can be chosen. For the k-fold, the gam around 0.367 and the sig2 around 0.497 can be chosen as the optimized parameter. For the leave one out, the gam and sig2 around 0.135 can be chosen.

For the cross validation, it splits the data into several folders, each folder is used for testing and others are used to train the model. Every data will be used for training and testing. The final error is averaged, which can remove the variance due to random split. For random split, the data is only split once, it may have a large variance.

If the number of folders is large, the model is trained on large dataset and validated on small dataset. If the number of folders is small, the model is trained on small dataset and validated on the large datasets. The k cannot be too small or too large, normally chosen from 5 to 10. Too large k value means that only a low number of sample combinations is possible, thus limiting the number of iterations. The large k value also has a high cost of computation. When the value of k is small, it is close to random split. And it will be close to leave one out method, when the k is extremely large. When choosing the value of k, we should also pay attention to the number of observations of training sets and test sets which should be large enough to make it statically meaningful.

1.3.3 Automatic parameter tuning

gam	Sig2	algorithm	runtime	cost
0.041731	0.57618	simplex	0.285126	0.03
1557.9885	0.4312933	simplex	0.229051	0.04
0.42633	0.02008	simplex	0.231352	0.03

1.3353	7.0508	simplex	0.218526	0.04
187.7772	0.05740829	gridsearch	0.616914	0.03
0.29537	0.023508	gridsearch	0.543505	0.03
0.040389	0.18481	gridsearch	0.508849	0.03
0.37984	0.26594	gridsearch	0.509724	0.04

Table 2. The gam, sig2, runtime and cost of each algorithm. Each algorithm has four runs.

The two automatic parameter tuning algorithm are tested, the runtime, cost and runtime are measured to compare the performance. For each algorithm, there are four runs. From the table 2, we can observe the game and sig2 differ a lot. The cost of different runs is close for different runs. For the run time, the gridsearch is about two times lower than simplex. Gridsearch is an exhaustive search method in a limited range, so it is slower. From the user guide, we can the tuning of parameters is conducted in two steps. The couple simulated annealing is used in first step to select the suitable parameters. And fine-tuning step is used to optimize the parameter based on the parameters selected from the first step, using the algorithms simplex or gridsearch.

For hyperparameter selection, it is non-convex problem, a lot of local minimums existing, no global minimum can be guaranteed. The CSA is global optimization methods, but we cannot know if the minimum it finds is global minimum or local minimum. So the final optimized parameters of different runs differ a lot.

1.3.4 Using ROC curves

The performance on the training set is always good, many methods find the best results based on the training set automatically. If we compare the ROC curve on the training set and choose the best model based on training set, we may face the issues of overfitting. The results of model will cause a large variance if we apply this model to other datasets.

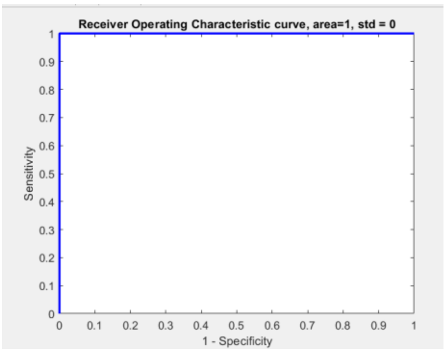


Figure 14. The roc curve based on the iris data.

The roc curve is generated based on tunned parameters and plotted on the test set. ROC curve can show the sensitivity and specificity of the model. AUC (area under curve) can be used to measure the performance, it is more useful when the area is great. From the iris dataset, the ROC curve is perfect, with high sensitivity and high specificity. And the AUC reaches the maximum area, indicating our model is perfect.

1.3.5 Bayesian framework

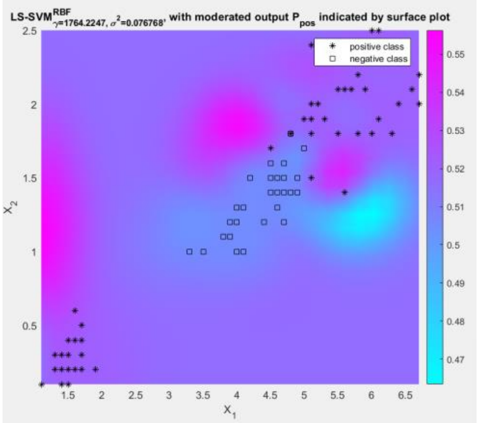
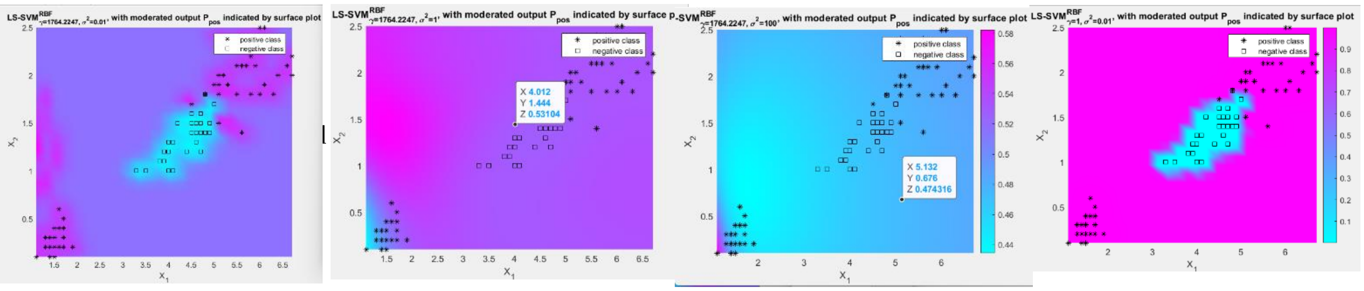


Figure 15. The plot generated from Bayesian framework/

The Bayesian framework is used to maximize the posterior probability to make the decision. The color indicates the possibility of belonging to positive class, we can see from figure 15, the purple color represents higher possibility of belonging the positive class and blue color represents the lower possibility of belonging the positive class.

Here I try different values of sig2 will make the very different output. With the tunned parameter, the performance is good. The data points with same class have close color, showing high probability of belonging to the same class. When the sig2 is changed, the performance is worse, the data points from different class have the similar color.



From this dataset, the training data contains 250 observations and 2 dimensions. The test data contains 1000 observations. Normally, the size of training sets is larger than the training sets. Otherwise, the testing errors would be very large. From the distribution of dataset, we can see the ideal decision boundary is not linear, but the linear kernel could also be used. I would like to try linear kernel, polynomial kernel and RBF kernel for this dataset.

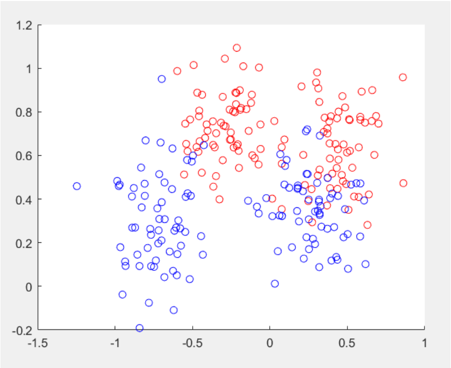


Figure 17. The distribution of Ripley data. The color represents different classes.

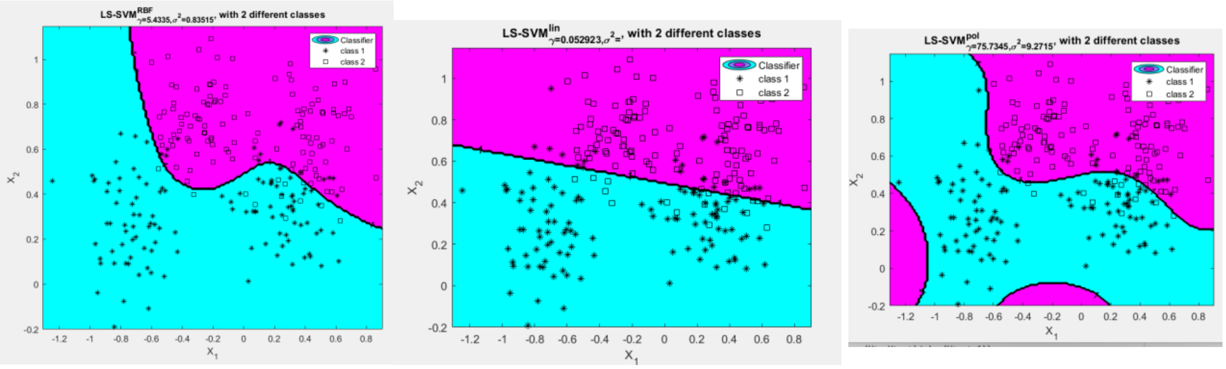


Figure 18. The classification with LS-SVM with different kernels. (From left to right: RBF, linear, polynomial kernel).

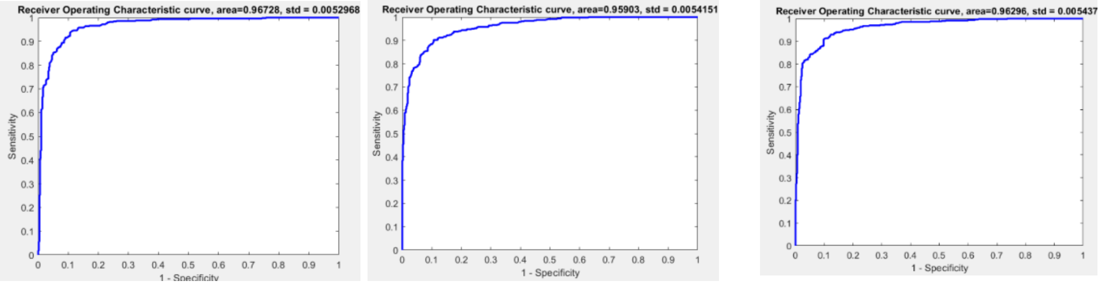


Figure 19. ROC plot of the classification with LS-SVM with different kernels. (From left to right: RBF, linear, polynomial).

For the RBF kernel the error rate is 0.0930 ,for the linear kernel, the error rate is 0.1050, for the poly_kernel, the error is 0.10. From the ROC curve, the RBF is slightly better (with larger AUC). Overall, the RBF performs better, however the difference is negligible. But I would like to choose the RBF kernel which can provide some knowledge about the structure of datasets. The accuracy of model is not perfect, only achieve 90%. I would like to resample the training set and testing sets which may achieve higher performance. And more variables can be considered to capture more features of different class and make better models.

For Diabetes data:

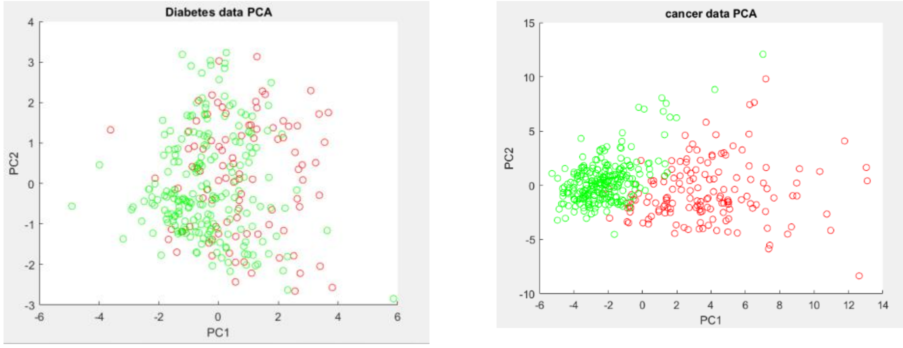


Figure 20. The distribution of diabetes data and cancer data projected on top 2PCs.

For diabetes data, it contains 8 dimensions. The data is projected on the first two PCs to visualize it. The data points are not separable from PCA plot, more complicated kernels are applied. The misclassification errors of poly_kernel, RBF_kernel and linear_kernel are 0.20833, 0.22619 and 0.23214 which are high. And the ROC plot does not perform well. I am not satisfied with the model, other methods may be better.

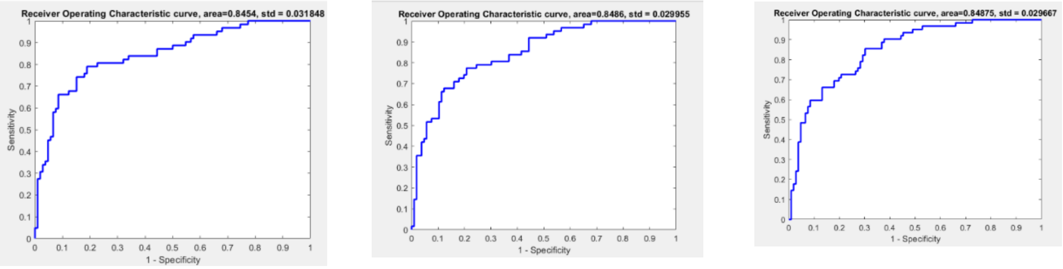


Figure 21. ROC plot of the classification of diabetes with LS-SVM with different kernels. (From left to right: linear, RBF, polynomial).

For the Wisconsin Breast Cancer dataset, the training set has the 400 observations and 30 dimensions. To visualize the data, the PCA is used to project the data into the first two principal components. We can observe the two classes are separated well on dataset. The linear kernel, polynomial kernel, RBF kernel are applied to the model, the misclassification error of each method is 0.0414,0.1497 and 0.01775. The ROC plots of each method are shown as below, the ROC plots are very close. I would like to choose RBF which has the lowest misclassification error and good performance of ROC plot. With the extremely low error, I am satisfied with this method.

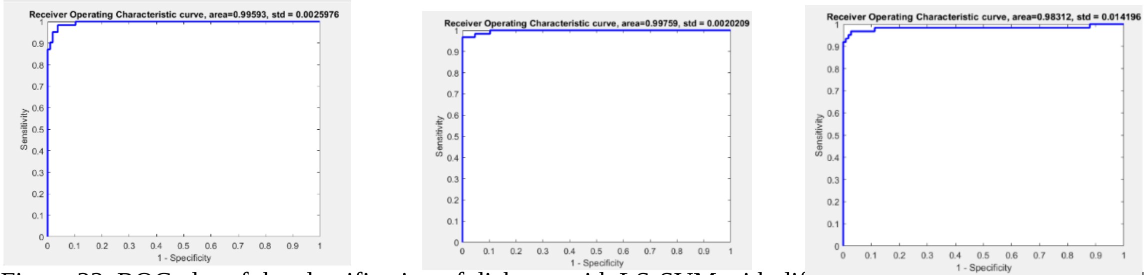


Figure 22. ROC plot of the classification of diabetes with LS-SVM with different kernels. (From left to right: linear, RBF , Polynomial)

Exercise Session 2: Function Estimation and Time Series Prediction

1.1 Support vector machine for function estimation

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?

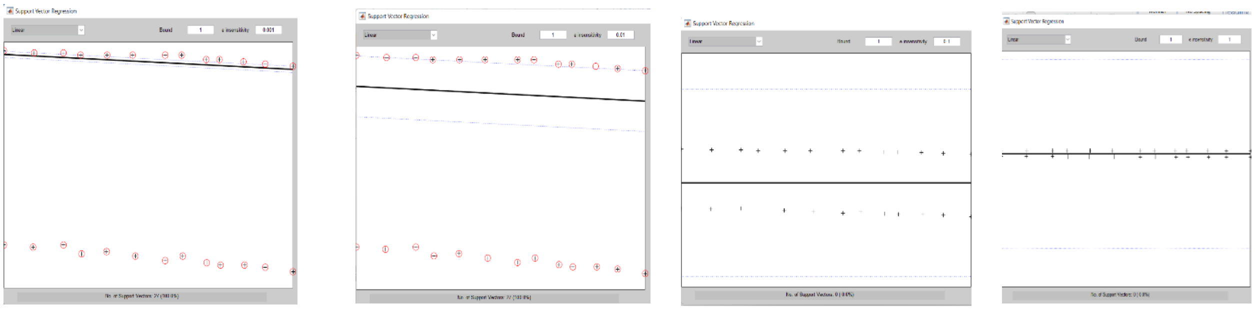

Support Vector Regression
Linear Bound 1 e insensitivity 0.001
Support Vector Regression
Linear Bound 1 e insensitivity 0.01
Support Vector Regression
Linear Bound 1 e insensitivity 0.1
Support Vector Regression
Linear Bound 1 e insensitivity 1

Figure 1. The SVM regression with linear kernel with fixed bound (bound =1) and different choices of e 0.001, 0.01,0.1,1 (left to right).

The figure 1 shows four plots with different e value for SVM regression with linear kernel. E represents the e-insensitivity regions, the points within this region can be tolerated and they are not penalized. As we can see, when the sensitivity increases, the margin expands. And the number of support vectors also decrease as more points are within the margin, less points are considered as support vectors.

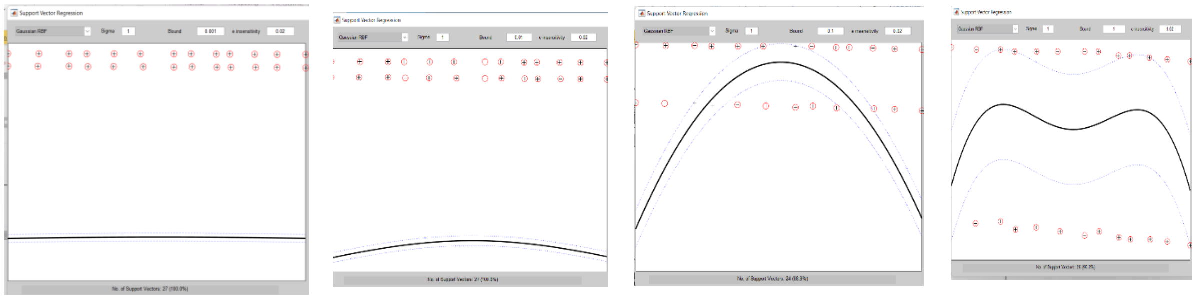

Support Vector Regression
Gaussian RBF Sigma 1 Bound 0.001 e insensitivity 0.02
Support Vector Regression
Gaussian RBF Sigma 1 Bound 0.01 e insensitivity 0.02
Support Vector Regression
Gaussian RBF Sigma 1 Bound 0.1 e insensitivity 0.02
Support Vector Regression
Gaussian RBF Sigma 1 Bound 1 e insensitivity 0.02

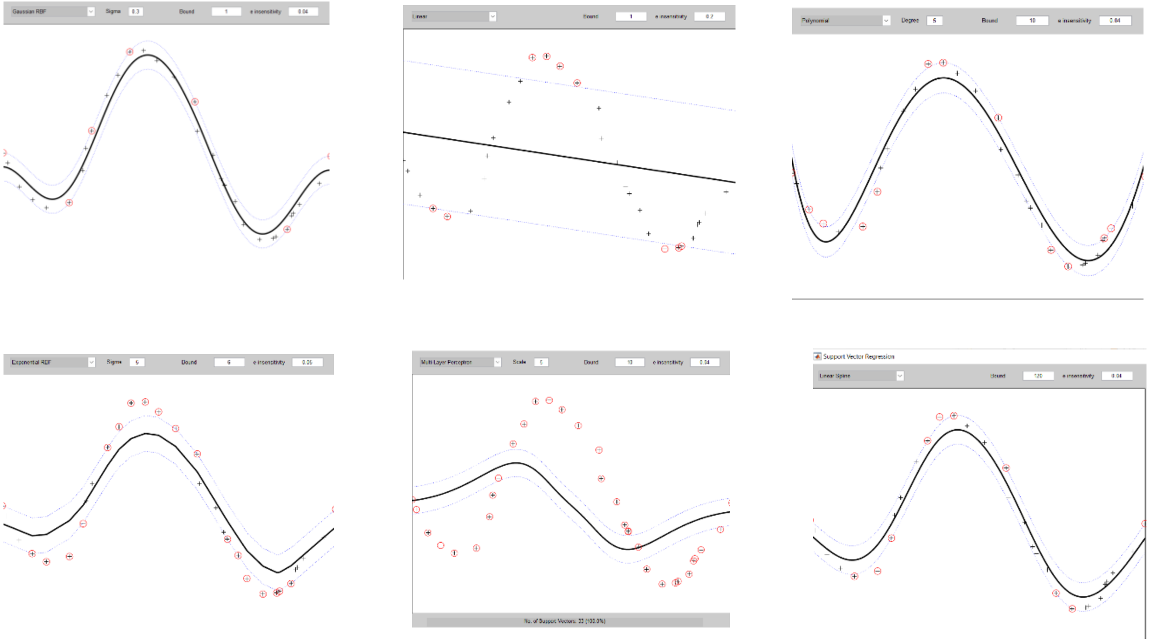
Figure 2. The SVM regression with gaussian RBF kernel for different choices of bound (0.001,0.01,0.1,1) (left to right) and fixed e (0.02).

The figure 2 shows the SVM regression with gaussian RBF kernel with different choices of bound. The bound values control the amount of regularization (smoothness of decision boundary). The increase of bound is associated with larger margin, more flexible decision boundary and better performance. When the bound is small enough, the decision boundary of gaussian RBF kernel is more smooth, close to linear kernel but extremely low performance.

The SVM solution is set of support vectors. Sparsity is from the data points located within the e-insensitivity region with zero Lagrange multiplier. The points from outside of e-insensitivity region are considered as support vectors.

Construct a more challenging dataset (around 20 data points). Which kernel is best

suited for your dataset? Motivate why.



Gaussian RBF Sigma 0.3 Bound 1 e insensitivity 0.04

Linear Bound 1 e insensitivity 0.2

Polynomial Degree 5 Bound 10 e insensitivity 0.04

Exponential RBF Sigma 5 Bound 6 e insensitivity 0.25

Multi Layer Perceptron Scale 5 Bound 10 e insensitivity 0.04

No. of Support Vectors: 33 (100.0%)

Support Vector Regression

Linear Spline Bound 120 e insensitivity 0.04

Figure 3. The SVM regression with different choices of kernel. (From left to right, top to bottom: gaussian kernel, linear kernel, polynomial, exponential RBF, multi-layer perceptron, linear spline)

Each kernel was tried, and the parameters of each kernel were selected and inspected manually. For my dataset, the linear kernel and multi-layer perceptron perform worst. And much more computation times was required for multi-layer perception. The exponential RBF acts slightly overfitting. The polynomial kernel is good, with a slight deviation for the peaks. The linear spline and gaussian RBF kernel perform best as they catch the non-linearity and the patterns.

In what respect is SVM regression different from classical least squares fit?

The largest difference between SVM regression and classical least square fit is different loss function they apply. For classical least square fit, L2 function is applied to find a line with the minimal distance from the observation. But for the SVM linear kernel, it uses L1 norm and the points inside the epsilon insensitivity region produce no error. The SVM linear kernel is less overfitted than classical least square fit.

1.2 A simple example: the sinc function

1.2.1 Regression of the sinc function

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).*

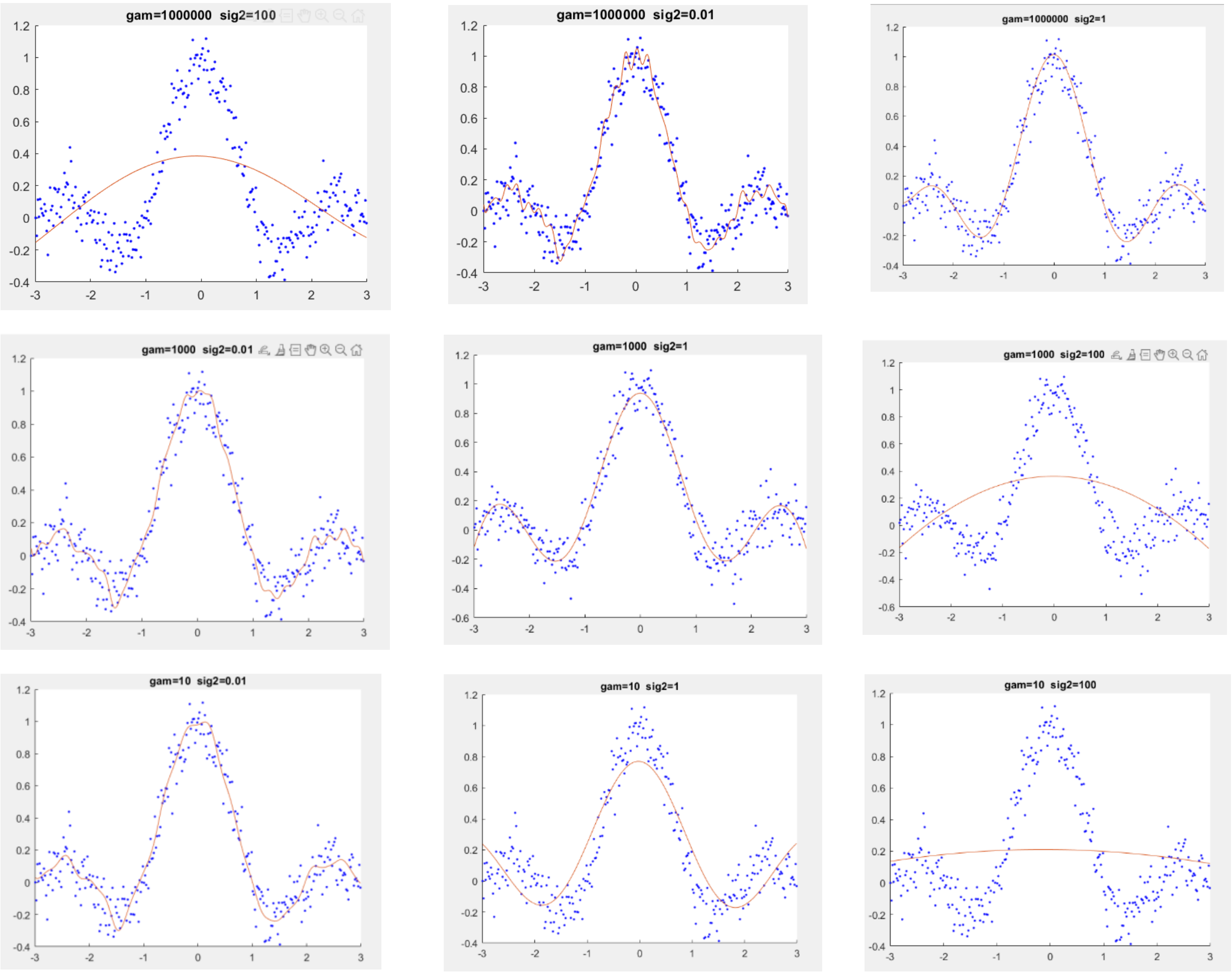


Figure 4. The visualization of different combinations of gam and sig2 of LS-SVM regression with RBF kernel of test set.

The different combinations of gam and sig2 are applied and models are trained with training sets and the results are visualized on the test set. The MSE can be seen for table 1.

gam	Sigma2	MSE
10	0.01	0.014
10	1	0.0330
10	100	0.1298
1000	0.01	0.012
1000	1	0.010
1000	100	0.1107
10^6	0.01	0.0121
10^6	1	0.0112
10^6	100	0.1042

Table 1. The MSE of different combinations of sigma2 and gam.

From the figure 4, two combinations (gam=1000 & sig2=1; gam=1000000&sigm2=1) perform best (low MSE and smooth fitted curve). Although some combinations (gam =1000000 & sig2= 0,01; gam=1000& sig2=0.01) also have low MSE but they are overfitted.

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

No, using the MSE as the criteria to select the best combining is not a convex problem, there are some local minimums, no overall minimums can be selected. The selection based on the MSE is not ideal, some issues such as overfitting cannot be avoided. For an ideal model, the model fits and model complexity should both be considered.

• *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	gam	runtime	MSE	cost	algorithm
0.916117261305	68794779.6532	1.377699	0.0105	1.012133e-02	Simplex
0.5636953026	151322.9189	1.309408	0.0105	1.012823e-02	Simplex
0.487238717	33579.3135	1.207901	0.0104	1.013539e-02	Simplex
0.36763259	2338.203	2.127339	0.0104	0.01016	Gridsearch
0.66866394091	1643586.2447	2.055999	0.0105	0.010126	Gridsearch
0.3661566	2781.3128	1.857574	0.0104	0.01015	Gridsearch

Table 2. The parameters tunning with different algorithms. (Leave one out is selected)

From table 2, we can observe the MSE of different algorithm for parameter tunning, The MSE for two algorithms has no significant difference, however the gridsearch algorithms spends more time to calculate.

1.2.2 Application of the Bayesian framework

Discuss in a schematic way how parameter tuning works using the Bayesian framework.

Illustrate this scheme by interpreting the function calls denoted above.

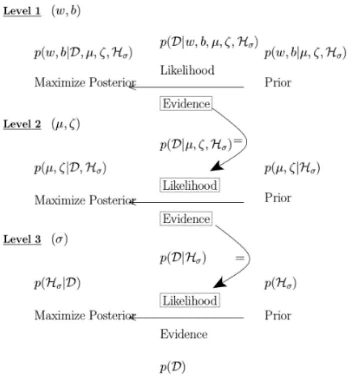

Level 1 (w, b)
$p(w, b|\mathcal{D}, \mu, \zeta, \mathcal{H}_\sigma)$
$p(\mathcal{D}|w, b, \mu, \zeta, \mathcal{H}_\sigma)$
$p(w, b|\mu, \zeta, \mathcal{H}_\sigma)$
Likelihood
Maximize Posterior
Prior
Evidence
Level 2 (μ, ζ)
$p(\mathcal{D}|\mu, \zeta, \mathcal{H}_\sigma)=$
$p(\mu, \zeta|\mathcal{D}, \mathcal{H}_\sigma)$
$p(\mu, \zeta|\mathcal{H}_\sigma)$
Likelihood
Maximize Posterior
Prior
Evidence
Level 3 (σ)
$p(\mathcal{D}|\mathcal{H}_\sigma)$ =
$p(\mathcal{H}_\sigma|\mathcal{D})$
$p(\mathcal{H}_\sigma)$
Likelihood
Maximize Posterior
Prior
Evidence
$p(\mathcal{D})$

Figure 5. The procedures of Bayesian inference levels (source: J.Suykens, Support Vector Machines: Methods and Applications chapter 4. Fig 4.1).

The figure 5 can indicate the three-level procedure of Bayesian framework, the evidence of each level become the likelihood of next level. The model is initialized with appropriate starting values, and for the first inference level, the alpha and b are optimized. The second level is to optimize the regularization parameter gam. For the third level, the kernel parameter sig2 is optimized.

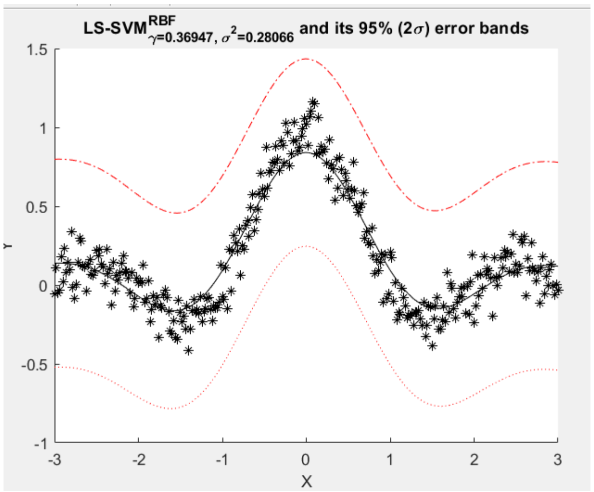


Figure 6. The 95% confidence intervals for regression with parameters optimized by Bayesian inference.

With the starting parameter values (sig2 = 0.4 and gam = 10), I obtained the final optimized parameters (gam=0.36947, sig2=0.28066). And the 95% confidence intervals can be obtained from Bayesian framework which is great.

1.3 Automatic Relevance Determination

- Visualize the results in a simple figure.

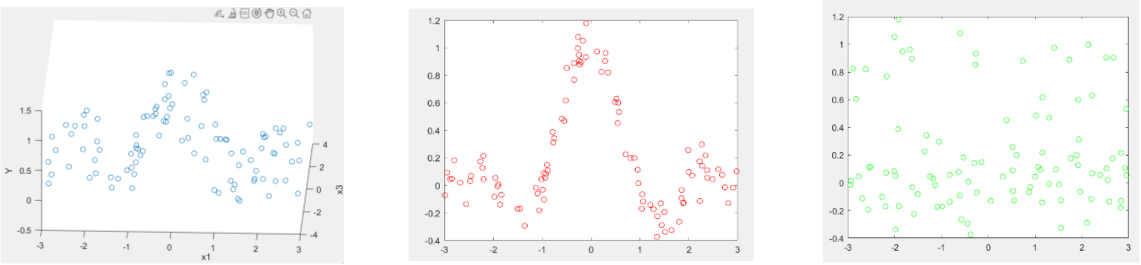


Figure 7. The 3D plot of simulated data(left). The projection of data on first dimension (middle) and the projection of data on third dimension (right).

Automatic relevance determination is applied to determine the subset of most relevant inputs to minimize the cost. The different dimensions are assigned to different weighting parameters. The input dimension with the largest optimal sig2 is removed in the step of backward selection.

The simulated dataset contains 100 rows and 3 dimensions. The tunned parameters from above are used. Among the three dimensions, the first and third dimension are selected. However, if I increase the times of run, the subset input may be different, the first dimension is always selected, indicating the large importance of first dimension.

The first dimension is considered as the most important dimension. From the figure 7, we can see the 3D plot of data, the data is very close to the sinc function. When projected on the first dimension (most important one) the sinc pattern becomes clearer. However, the projection of data on the third dimension shows no pattern.

How can you do input selection in a similar way using the cross-validation function

instead of the Bayesian framework.

As describing before, the Bayesian framework is used to select the input dimensions to minimize the cost associated with the third level of Bayesian inference. The cross validation can be used to compare the performance of models with different combination of variables. The error on the validation sets can be calculated as the measure of generalization performance. The sig2 can be selected from minimizing the validation error.

1.4 Robust regression

Visualize and discuss the results. Compare the non-robust version with the robust

version. Do you spot any differences?

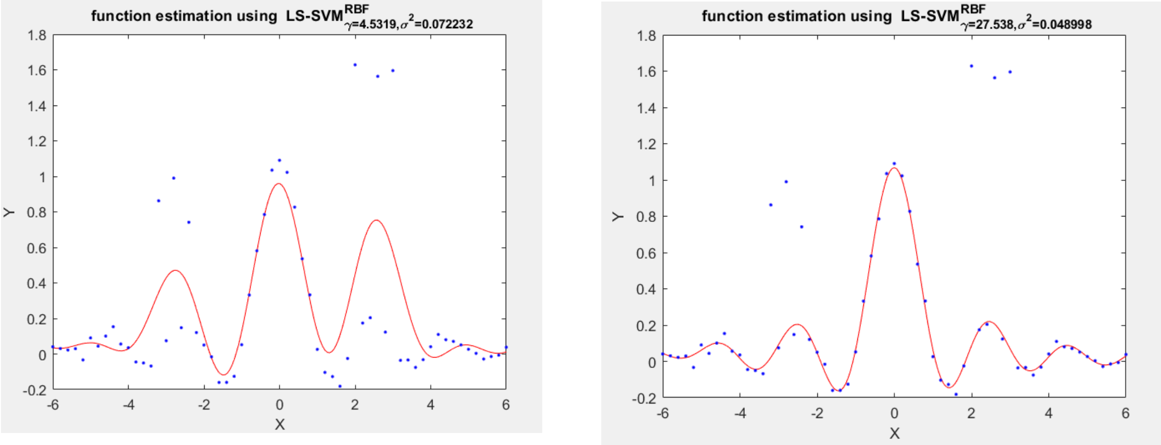


Figure 8. The function estimation of robust regression (right) and non-robust regression (left).

From figure 8, we can observe the difference of robust version and non-robust version. The non-robust version is affected by the outliers, making the regression lines biased. The robust version makes better performance, ignoring the outliers.

Why in this case is the mean absolute error ('mae') preferred over the classical mean

squared error ('mse')?

Because MSE is more highly affected by outliers. The outliers always have a large distance from other points and the squared distance between outliers and fitted line will be extremely larger, putting more weights on the outliers when doing regression. Thus, the MAE is preferred.

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.

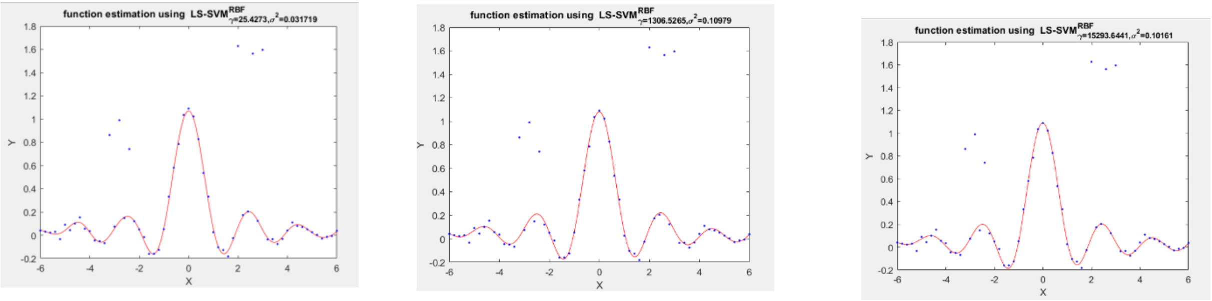


Figure 9. The robust function estimated with different weighting function (From left to right : whampel, wlogistic, wmyriad).

There is no significant difference between the function estimation of three different weighting function, they all perform well.

2 Homework problems

2.1 Introduction: time series prediction

Time series prediction

As indicated numerous times before, the parameters gam and sig2 can be optimized

using cross validation. In the same way, one can optimize order as a parameter. Define

a strategy to tune these 3 parameters.

In order to optimize the gam, sig2 and also the order, the simplex is used to tune the sig2 and gam with 10-fold cross validation. A sequence of orders (5:5:100) are used for test, the MSE and MAE are used to evaluate the performance separately.

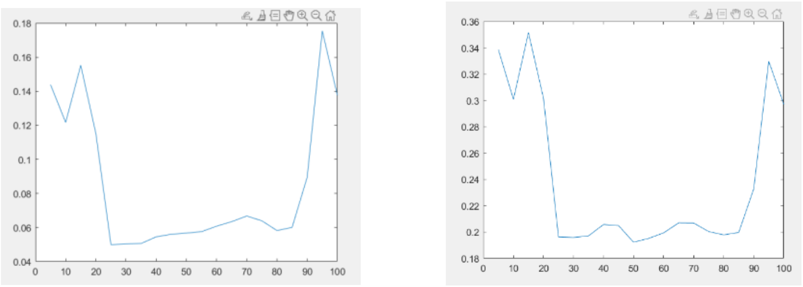


Figure 10. The performance of different choices of orders with tunned gam and sig2. (Left: mse, right mae)

From figure 10, we can observe ideal value of order is around 25 based on mse. The performance measured by mae is close to mse, the ideal value is around 25 or 50 for choosing the mae as the criteria for measuring performance.

- *Do time series prediction using the optimized parameter settings. Visualize your*

results. Discuss.

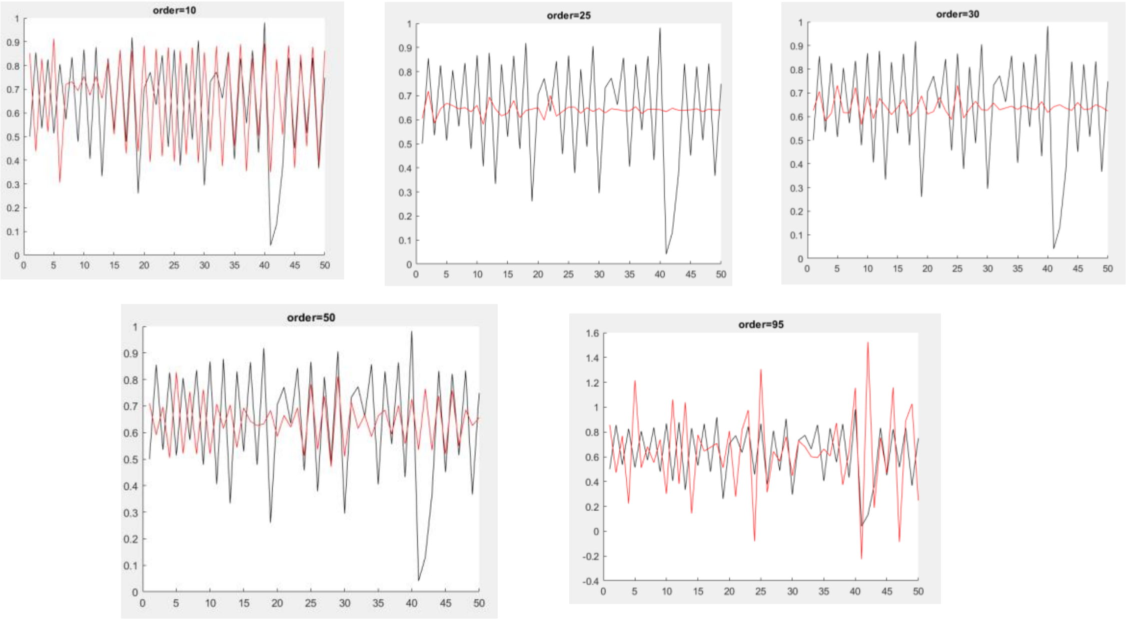


Figure 11. The prediction and real fit of time series prediction. Read line is the prediction and black line is the real line.

To visualize the prediction, we can see final fits of different value of order. The orders with 25,30,50 are considered as optimized parameters of gam, sig2 and order. The orders with value of 10 and 95 are considered as very poor parameter. The fitted curves are not ideal, as they do not capture the dramatic changes of different time points, for example, when the order is 25, the fitted curve is relatively flat, showing limited information of prediction. The prediction based on the condition that the value of order is 50 can fit some peaks, however some time points they use the reverse peak pattern (time 0-10). The time series prediction based on optimized parameter is not ideal.

2.3 Santa Fe dataset

Does order = 50 for the utilized auto-regressive model sounds like a good choice?

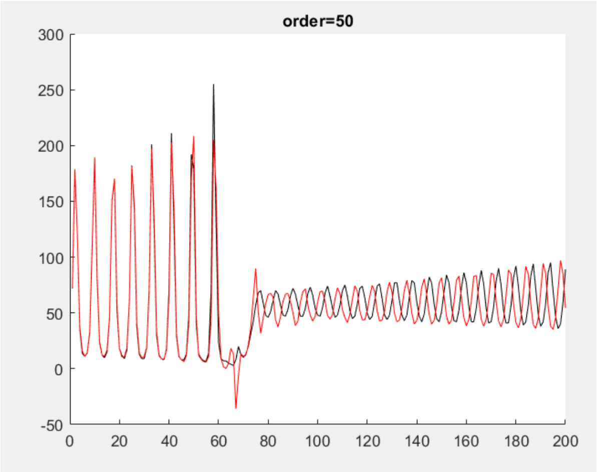


Figure 12. The prediction and real fit of the Santa Fe dataset with order is 50.

For this dataset, the data contains 1000 time points. In order to tune the parameters, the “mae” is used rather than mse due to dramatic changes of value for adjacent time points. The 200 time points are used to test the performance of prediction when the order is 50. From the figure, we can see before 60, the prediction is fitted with real data perfectly, however after the 60, the prediction is deviated, reversing the positions of peaks.

• *Would it be sensible to use the performance of this recurrent prediction on the validation set to optimize hyperparameters and the model order?*

We can use the validation set to optimize the parameters, however, when we split the data into training sets, validation sets and test sets, we should make sure the time points of the split data are continuous. It is also essential to select a good size of validation sets. The variance could be very large, when the range of time is large.

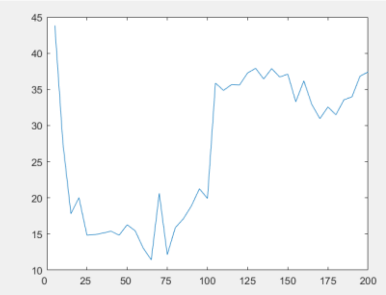


Figure 13. The mae of different orders. The x-axis is the order and the y-axis is the mae.

From the figure, we can observe the global minimum around 65. The order 60,65,75 are selected to make the prediction. When the order is 60, it makes the best prediction.

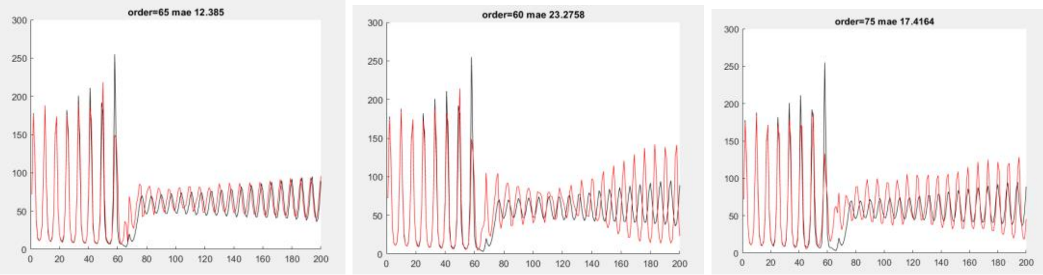


Figure 14. The prediction and real fit of the Santa Fe dataset with different choices of order. (From left to right : 65,60 and 75).

Exercise Session 3: Unsupervised Learning and Large Scale Problems

1 Exercises

1.1 Kernel principal component analysis

• Describe how you can do denoising using PCA. Describe what happens with the denoising if you increase the number of principal components

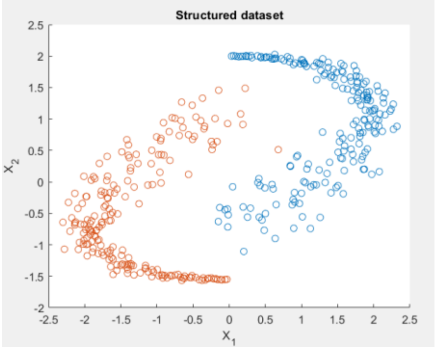


Figure 1. The distribution of original dataset. Color represents different class.

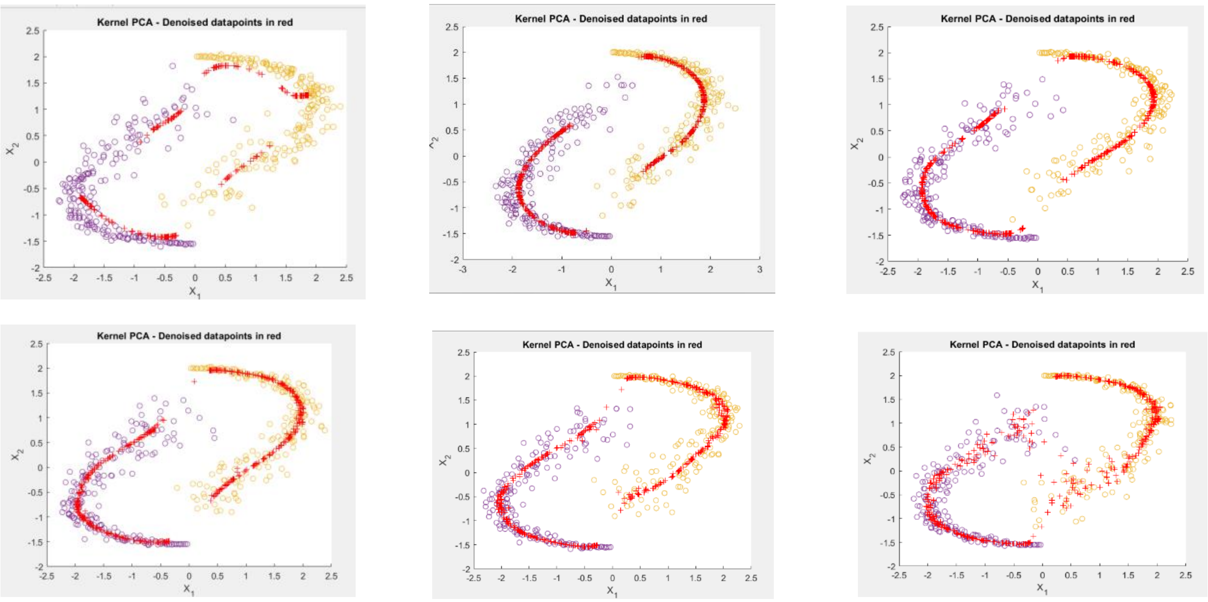


Figure 2. The kernel PCA with different number of PCs(From left to right :PCs=2,4,6, second row: 8,10,12)

From the figure 2, we can see the when the PCs is 6 or 8, the results of denoising are good. However, when the number of extracted components keeps increasing, the model starts to be overfitted.

For PCA, the main idea of PCA is to re-construct the new dimensions which is called as principal components. Using the suitable number of PCs which can explain the majority of variance. hen projected on the new dimensions, the most information can be kept. The dimensions with low eigenvalues can be considered as noise.

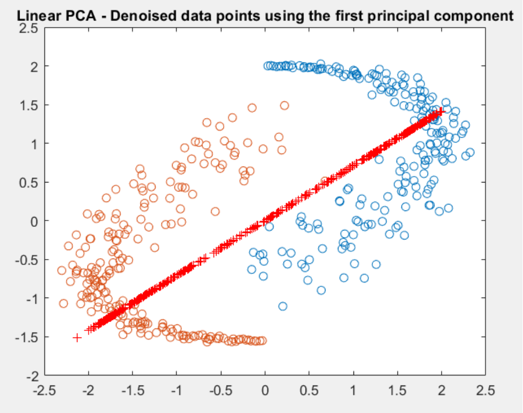
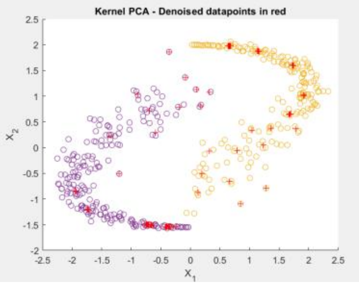


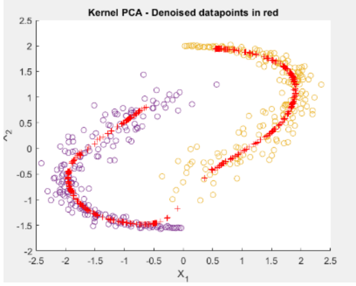
Figure 3. The classification with linear PCA.

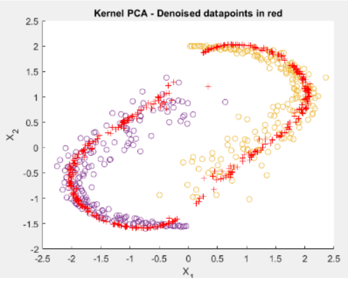
For the linear PCA, the new PCs are the linear combination of the original variables. The kernel PCA is used for nonlinear decision boundary. The maximum number of PCs is equal to the number of original dimensions, here the maximum number of PCs is 2. For this situation, the linear PCA is poor, as the decision boundary is not linear, even we introduce the second PC, the clustering is still not ideal.

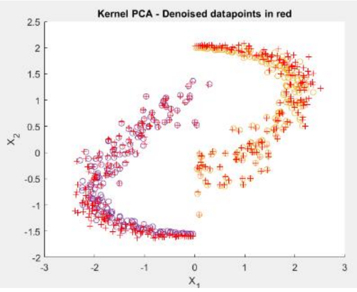
But for kernel PCA, the kernel function is used to project dataset into a higher dimensional feature space, where it is linearly separable. The distances between the datapoints and the Kernel matrix are computed. The eigen-analysis based on the kernel matrix is done. The maximum PCs can be the dimension of kernel matrix which is the 400 here. The kernel PCA can introduce the non-linearity which performs very well in this dataset.

•*For the dataset at hand, propose a technique to tune the number of components, the hyperparameter and the kernel parameters*









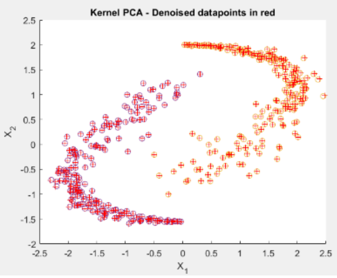


Figure 4. The kernel PCA with different kernel parameter sig2 (From left to right : 0.01,0.4,1,10,100)

The different sig2 are chosen to test the classification performance of kernel PCA. From the figure 4, we can observe when the sig2 increase from 0.01, the data points are well denoised. When the sig2 keep growing, the issues of overfitting occur, more noises are accounted in the model, leading to poor denoising performance.

$$\tilde{x} = h(z) \qquad \min \sum_{k=1}^{N} \|x_k - h(z_k)\|_2^2$$

Formula 1

Reconstruction error is applied as the criteria for the performance. The reconstruction is calculated as formula 1.

Sig2|| nc	2	6	24	50
0.01	95.6653	87.1771	43.0724	30.8794
0.1	174.6881	122.67	30.5951	7.7942
1	256.37	58.2828	3.1855	0.1607
10	635.7961	22.9981	0.1074	700.3208

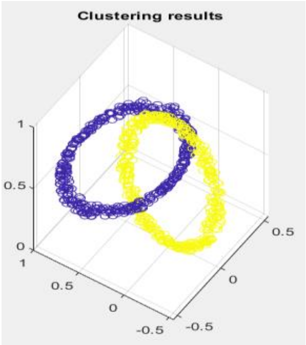
Table 1. The reconstruction errors of each combination of sig2 and number of PCs.

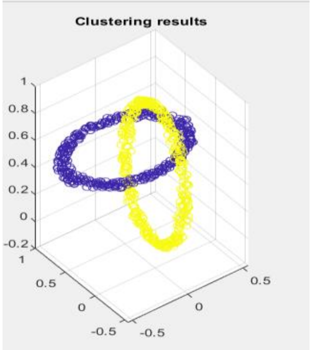
From the table 1, we can when the nc is 24 and the sig2 is 10, the reconstruction error is lowest. The ideal range of nc is from 24 to 50 and the ideal range of sig2 is around 1.

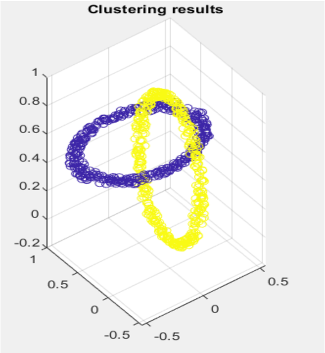
1.2 Spectral clustering (optional)

For spectral clustering, the affinity matrix (kernel matrix) is obtained firstly which represents the similarity of each pair of data points. And the degree matrix is also generated.
The Laplacian matrix is obtained by rescaling the kernel matrix. The second and third largest eigenvalues/vectors can be obtained using Lanczos. The clustering is conducted on the subspace spanned by second and the third largest eigenvectors.

For the clustering, it is unsupervised learning method, no labels of datapoints are known. The data points who are similar (short distance) are clustered together. For the classification it is supervised learning, the labels are known before the classification, the misclassification can be measured. The goal is to achieve a good classification with low misclassification error. Six sig2s are used here (0.001 ,0.005 ,0.01 ,0.1 ,1 ,10). When the sig2 is from 0.001 to 0.01, the two rings can be separated well, reaching good clustering results.







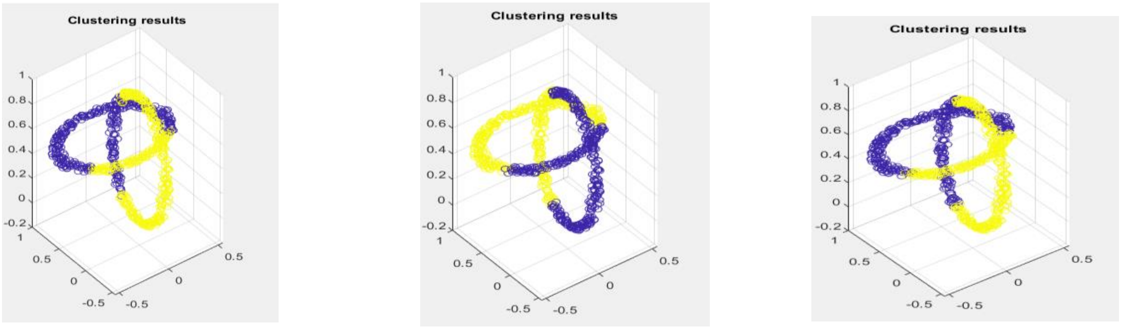


Figure 5. The clustering resulting with different sig2 value is shown in 3D space, the colour represents different clusters. (From left to right, top to bottom : 0.001 ,0.005 ,0.01 ,0.1 ,1 ,10).

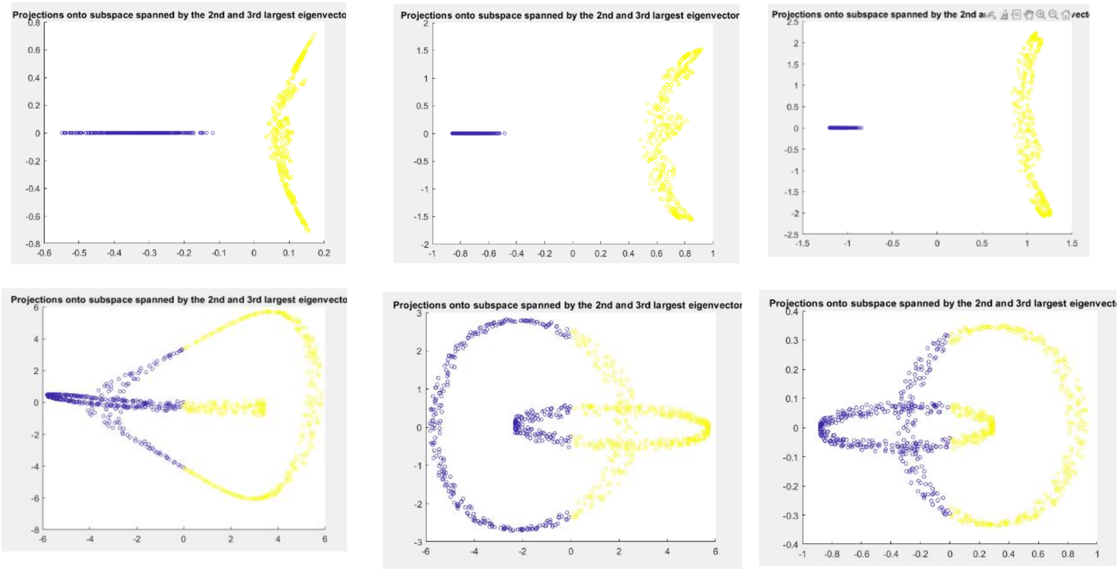


Figure 6. The projection of datasets onto the subspace by 2nd and 3rd largest eigenvalues. The colour represents different clusters (From left to right, top to bottom: 0.001 ,0.005 ,0.01 ,0.1 ,1 ,10).

1.3 Fixed-size LS-SVM

In which setting would one be interested in solving a model in the primal? In which

cases are a solution in the dual more advantageous?

For the primal problem, the number of parameters which are required to be optimized is equal to number of dimensions of original datasets. For the dual problem, the number of parameters which are required to be optimized is equal to the number of all data points. The data with low dimensions but high number of instances is suitable for solving in primal. In contrast, the data with high number of dimensions but small number of instances are suitable for solving in dual.

Mapping to feature space could be expensive for computing if the dimensions are extremely high.

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

hat subset the algorithm converges.

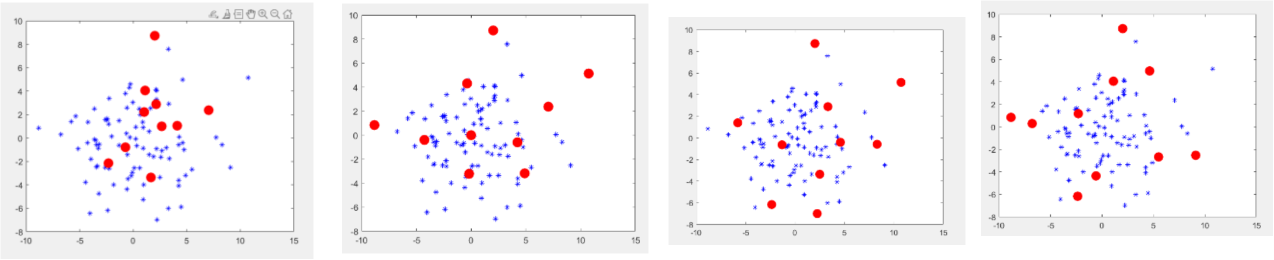


Figure 7. The subset of support vector with different sig2 (From left to right: 0.01, 0.1,1,10)

The influence of different value of sig2 on the subset selection is displayed here. With the higher value of sig2, the sub selected points in the space are much more widely distributed. The data points can represent the features of each region. The subset algorithms will converge to the data points can cover the while regions and these data points represent best to build a good model in the feature space.

• *Run fslssvm script.m. Compare the results of fixed-size LS-SVM to `0-approximation*

in terms of test errors, number of support vectors and computational time.

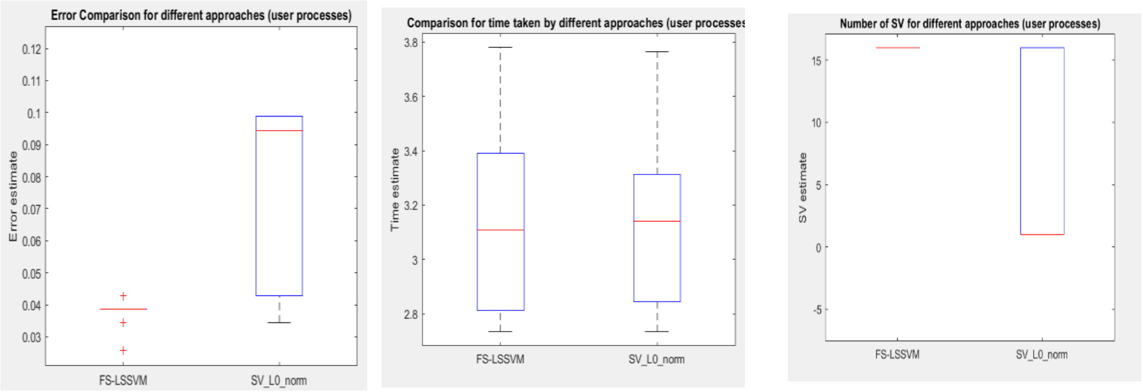


Figure 8. The error, time taken and the number of support vector for the Fixed size LSSVM and L0_norm approximation method for top 700 data points of shuttle.

From the figure, the run time of FS-LSSVM and SV_L0 norm is very close. For the number of support vectors, the FS-LSSVM has large number of support vector. The SV_L0 norm tends to have lower number of support vector, resulting in more spare result. For the estimation of error, the SV_L0_norm has higher error and the variance of error of SV_L0_norm is also very large.

2 Homework problems

2.1 Kernel principal component analysis

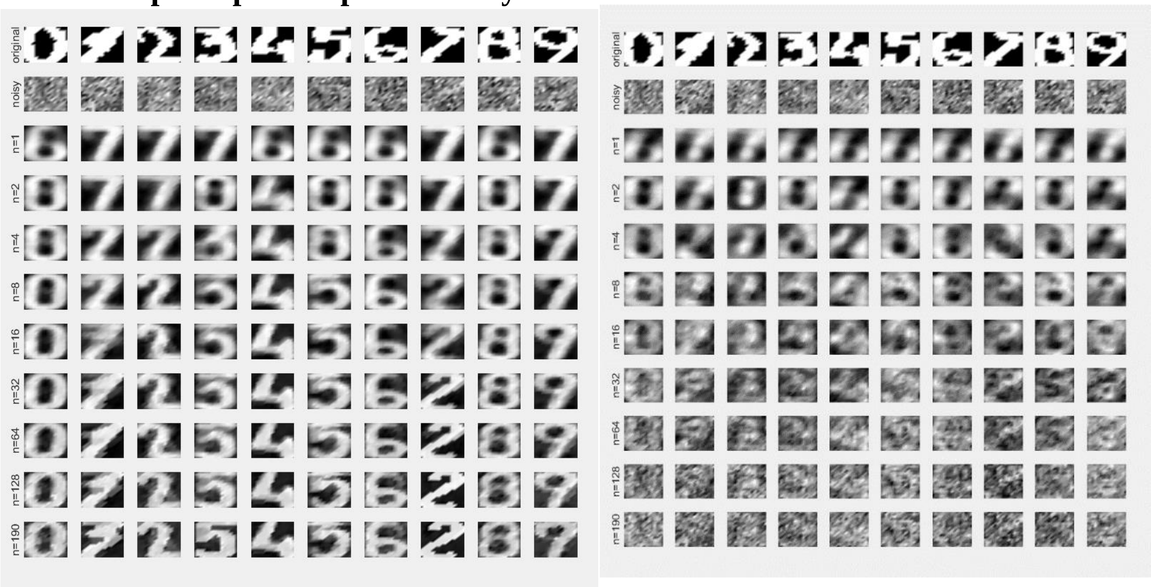


Figure 9. The digital handwriting image denoising with linear PCA and PCA kernel with RBF kernel.

The figures show the digital handwriting image denoising with linear PCA and kernel PCA with RBF kernel. The noise factor is selected as 1. We can see the numbers with a lot of noise from the second row of the figures, the numbers are impossible to be recognized. When the extracted components of kernel PCA is larger than 32, the results of denoising are very good, the pattern of number can be clearly recognized. But for linear PCA, when the extracted components are 2,4 or 8, the denoising results are the best comparing with other number of extracted components, but the numbers are still not clear enough to be recognized .

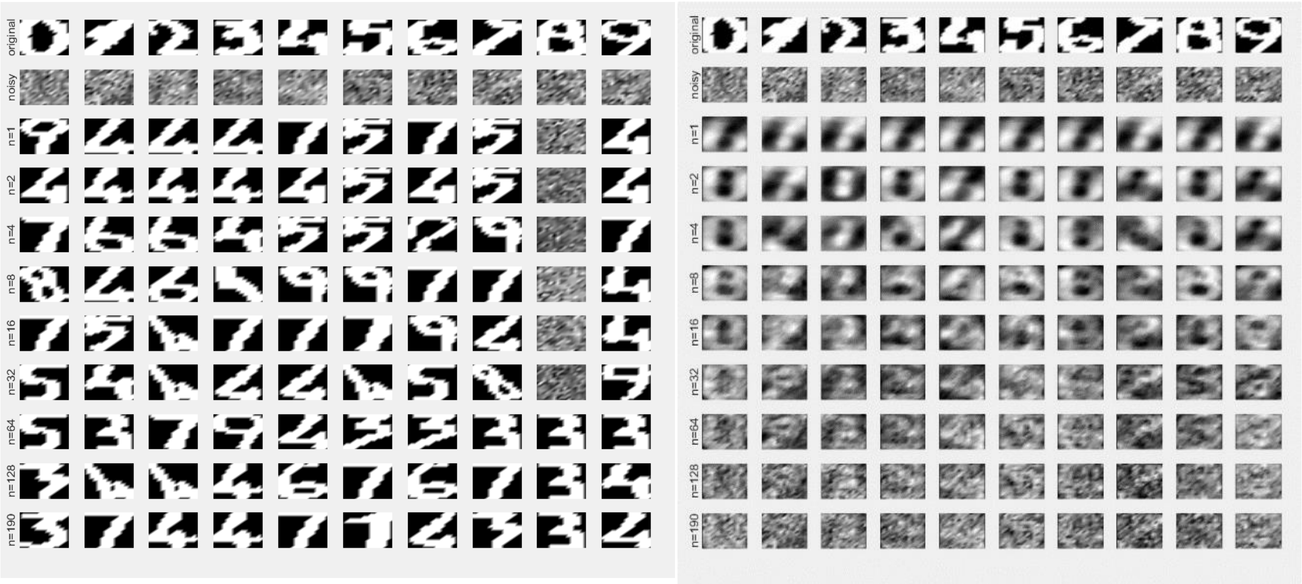


Figure 10. Denoising the handwriting digital image with different sigma factor. (Left:0.01 , and right:100).

The sigma factors 0.01 and 100 are applied to denoise the digital handwriting data. In this situation, the small sigma factor works well to denoise the data comparing with the large sigma.

The extremely large sigma did not improve the reorganization of number a lot, the image is still very difficult to be recognized. With the much smaller sigma, the image is denoised very well, the patten of number is clear. However, some important original data is lost, the number is not correct.

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?

The reconstruction error is received from the “recerrors” of kpca function. The input parameter, ‘sig2’,’test’ are adjusted to set several choices of the value of sig2. The error of 10 images is added together to represent the overall error. The selected value of sig2 is 10.^power. The power is -5:1:5. From the plot of errors, we can observe the increase of sig2 can reduce the errors.

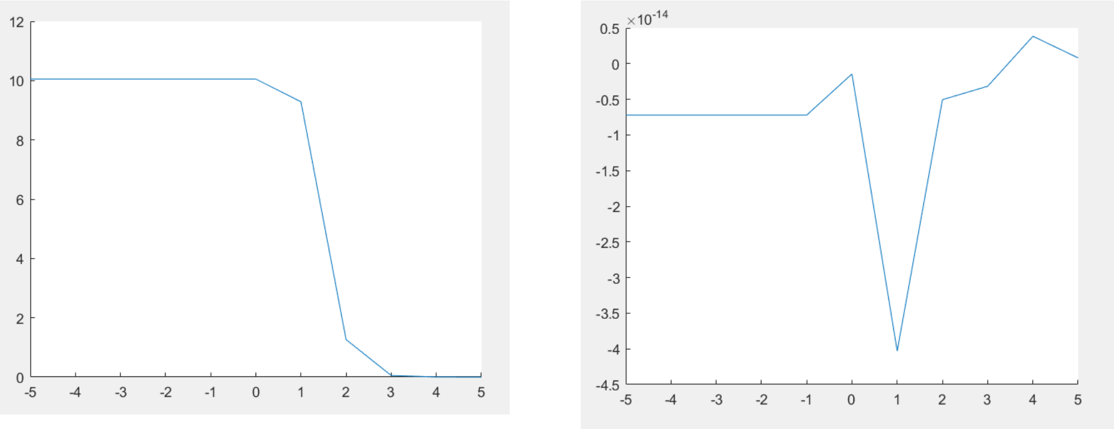


Figure 11. The reconstruction errors with increase of sig2. The x-axis is the log(sig2) and the y axis is the overall reconstruction error.

The reconstruction errors based on Xtest1 and Xtest2 can be seen from the figure 11. We can see with the increase of power of sig2, the reconstruction errors reduce to 0 for test1, however for Xtest2, there is one minimum which power is equal to 1. Choosing the optimized sig2 to make reconstructed images based on test2, we can observe the reconstruction is good when the number of extracted component is larger than 16. All the numbers are corrected reconstructed.

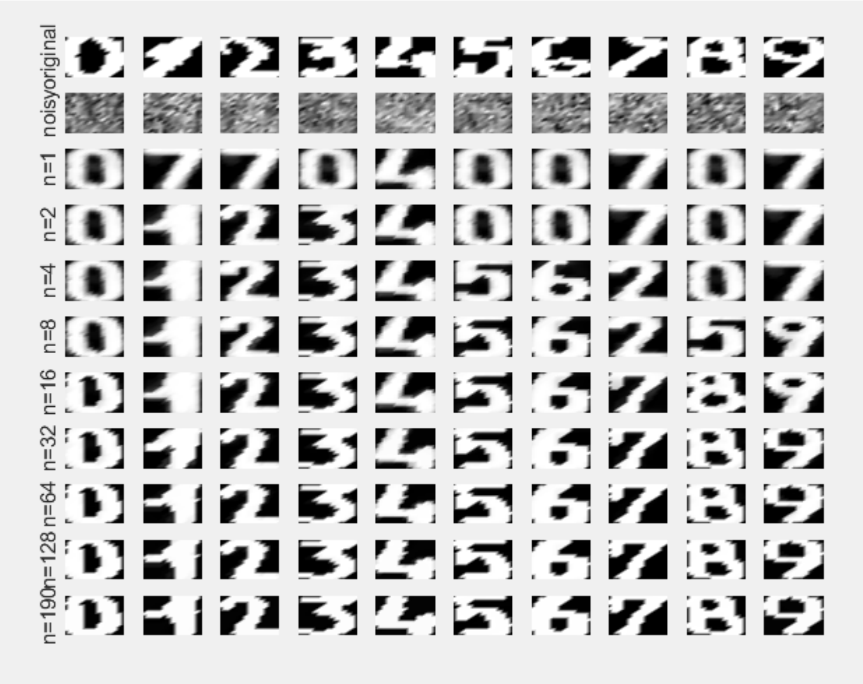


Figure 12. Denoising the handwriting digital image with different sigma=16 and validated on test 2.

2.2 Fixed-size LS-SVM

2.2.1 Shuttle (statlog)

For the shuttle dataset, there are 58000 observations belonging to 7 classes. There are 9 attributes. The classification performance is expected to reach a high accuracy as the number of observations is ridiculously huge. The distribution of 7 classes can seen from the histogram, most of the observation belongs to the class1.Class 2,6,7 have a relatively smaller number of observation points.

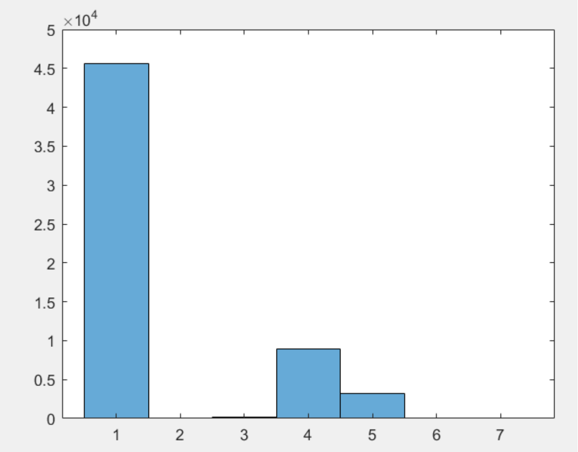


Figure 13. The distribution of seven class of shuttle dataset.

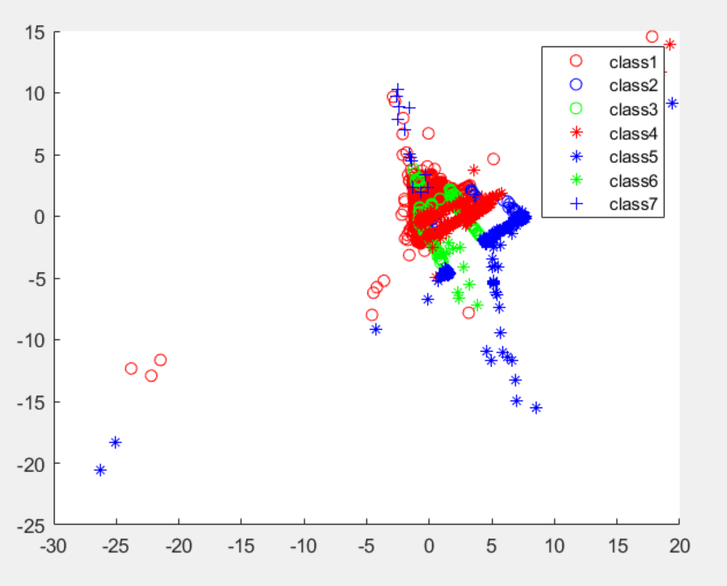


Figure 14. The projection of data on two PCs.

The projection of data on two PCs can be found from figure 14. The classes of observation are not easy to be separated. The linear kernel is used make the classification based on full dataset. The comparison between the fixed size LS-SVM and L0_norm approximation methods is shown as figure 15. For the running time, two methods have close performance. For the error estimation, we can see the L0_norm approximation has larger value and larger variance (the mean difference is 0.005). For the number of support vectors, the L0_norm maintains large number of support vectors, showing the sparsity estimated by the L0_norm approximation method.

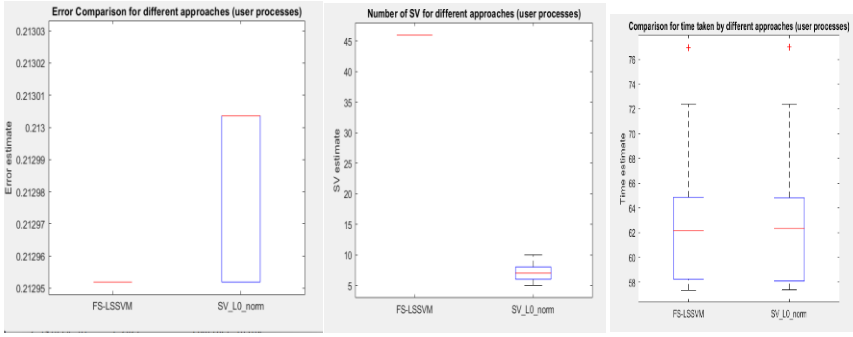


Figure 15. The error, number of support vector and run time of Fixed size LSSVM and SV_l0 approximation method with linear kernel of shuttle dataset.

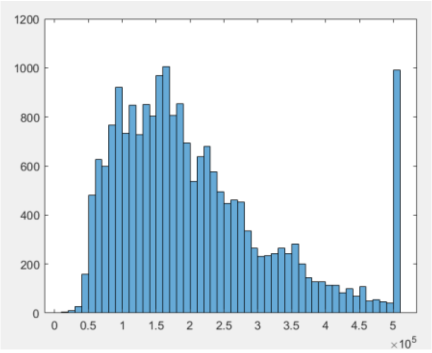


Figure 16. The distribution of ln (median house value) from California dataset.

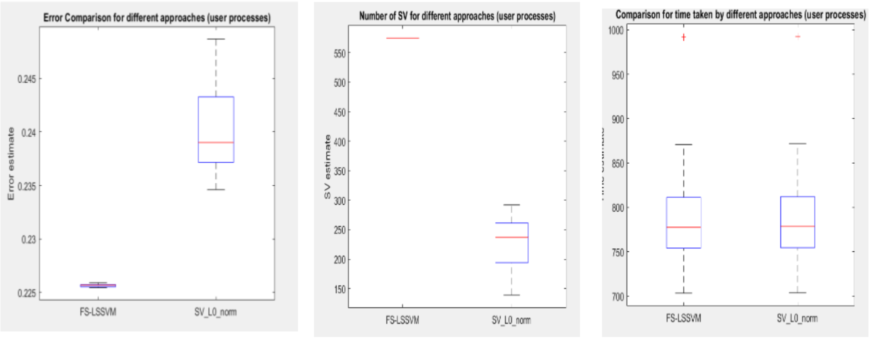


Figure 17. The error, number of support vector and run time of Fixed size LSSVM and SV_l0 approximation method with RBF kernel of California dataset.

For the California dataset, it totally includes 20640 data points with 9 attributes, containing 8 independent variables and one dependent variable (ln(median house value)). The distribution of ln(median house value) can be seen from figure 16. All variables are continuous, the goal is to do good function estimation, making a good regression. The RBF kernel is used for fixed size LSSVM and L0 approximation method. From the result plots, we can observe that the computation time is close for two methods. And the L0_norm approximation method is sparer with the cost of high errors. The fixed size LSSVM has very smaller errors, the different between the mean of errors from two method is around 0.0175.