Katholieke
Universiteit
Leuven

**Master of
Artificial Intelligence**

SUPPORT VECTOR MACHINES: METHODS AND APPLICATIONS

Course Report

Daichen Li (r0867950)

Academic year 2022–2023

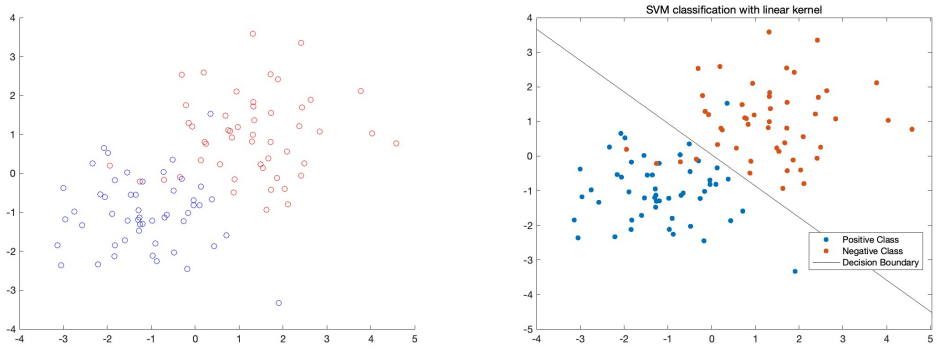
Contents

1 Classification

1.1 Exercises

1.1.1 A simple example: two Gaussians

Depending on the distribution of the data, we may choose to use a linear classifier to separate positive and negative data. In this example, we can use the perceptron algorithm to obtain an optimal decision boundary that minimizes the number of misclassified points separating positive and negative data. This linear classifier is optimal because it correctly classifies as many data points as possible and misclassifies the fewest number of data points.



(a) Scatter plot of the dataset (b) Decision Boundary Diagram

Figure 1: Data Distribution

The first output Figure 1a is a scatter plot of the data set, where the red circles represent positive data and the blue circles represent negative data. The second Figure 1b is the decision boundary obtained by the classifier, that is, the linear boundary separating positive and negative data.

1.1.2 Support vector machine classifier

Support vectors are the training sample data points used when training the SVM. The support vectors are the data points that lie closest to the boundary of the hyperplane. In a classifier, when classifying a new data point, only the support vectors contribute to the classification result. As data points are adjusted or added to the dataset, the location and number of support vectors may change, and their importance may change accordingly. The support vector importance magnitude is represented by the size of the great circle in the online application.

As more data points are added to the dataset, if these data points are classified correctly, they will increase the support for the classification hyperplane, making the classification hyperplane more stable. Conversely, if these data points are misclassified, they will have an impact on the position of the classification hyperplane and may lead to a shift or misclassification of the classification hyperplane. In this online application, we can change the classification results by changing the values of the regularization hyperparameter C and the kernel parameter sigma.

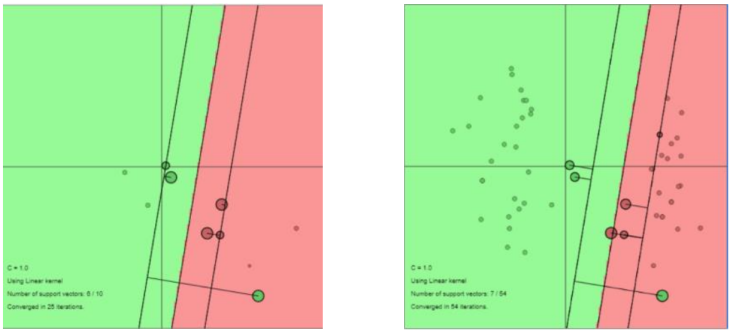
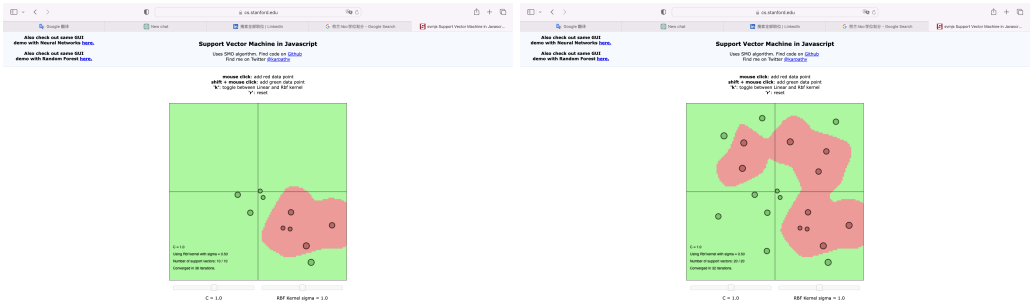


Figure 2: SVM Classifier with Linear Kernel: Add More Data Points

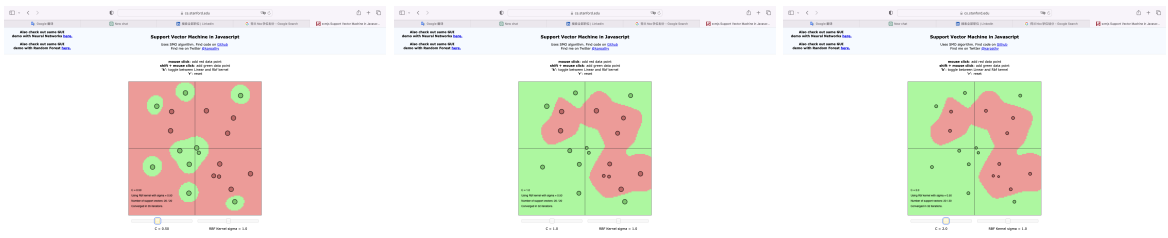
As we add more data points to the dataset, as shown in Figure 2 and 3, we can see that the classification hyperplane is doing its best to separate the data points correctly. Some data points become support vectors, which are the points closest to the classification boundary, and the location and number of these points affect the location and shape of the classification hyperplane.



(a) (b)

Figure 3: The SVM Classification with RBF Kernel: Add More Data Points

When considering the RBF kernel, we observe that a low value of C emphasizes the misclassification errors, prioritizing accurate classification. On the other hand, a large value of C results in a smoother decision boundary, where the importance of support vectors becomes more balanced and evenly distributed, as shown in Figure 4.



(a) C=0.5 (b) C=1 (c) C=2

Figure 4: The SVM classifier with RBF kernel for different choices of C parameter.

By analyzing Figure 5, it is evident that as the value of the C parameter decreases, the margin expands, and a larger number of support vectors are considered. The C parameter, serving as the regularization parameter, determines the cost associated with misclassification on the training data. When C is larger, a lower error rate is permitted. This trade-off controls the balance between the cost of misclassification and the margin of the decision function. A larger value of C increases the cost of misclassification, allowing for less error but resulting in a narrower margin.

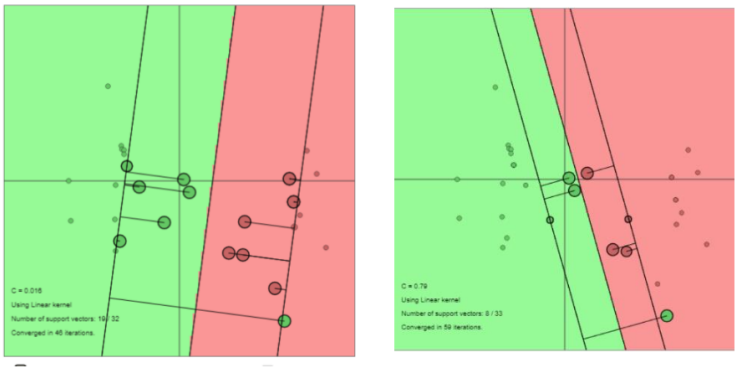
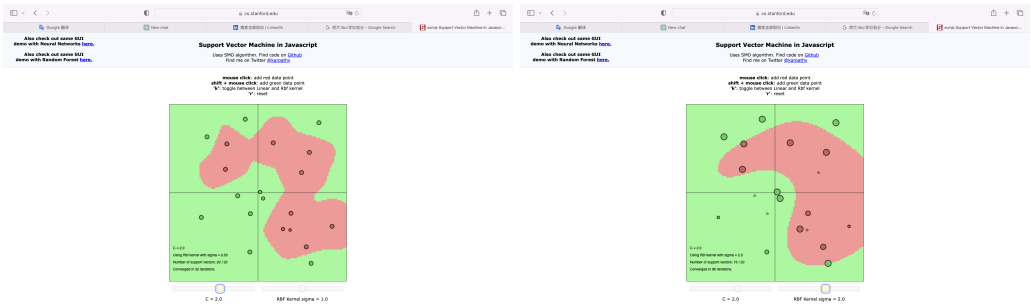


Figure 5: The SVM Classifier with Linear Kernel for Different Choices of C Parameter

The kernel parameter sigma is used to adjust the shape of the RBF kernel. As shown in Figure 6, a larger sigma value will result in a smoother decision boundary, and a smaller sigma value will result in a sharper decision boundary. When sigma is very large, the classification boundary becomes very smooth, because the RBF kernel function assigns very little weight to the points that are far away, making these points have little influence on the classification decision.



(a) sigma=1 (b) sigma=2

Figure 6: Different Choices of Sigma

1.1.3 Least-squares support vector machine classifier

When evaluating the model on the test sets, the misclassification rate is computed as an indicator of performance. It has been observed that as the degree increases, the misclassification error tends to decrease. Notably, when the degree is set to 0, the error rate starts at zero.

However, it is worth noting that using a large degree may lead to the problem of overfitting. Overfitting occurs when the model becomes too complex and starts to fit the training data too closely, as shown in Figure 7, resulting in reduced generalization ability on unseen data.

To mitigate the risk of overfitting, alternative approaches can be considered. For instance, regularization techniques, such as L1 or L2 regularization, can be applied to control the complexity of the model and prevent it from over-adapting to the training data.

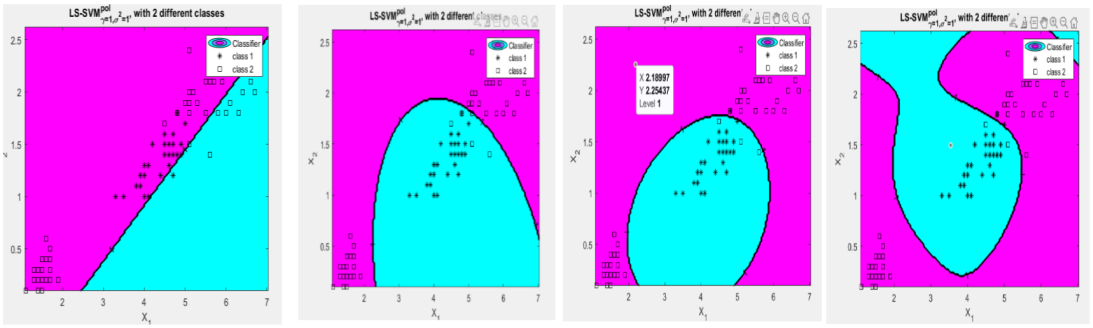
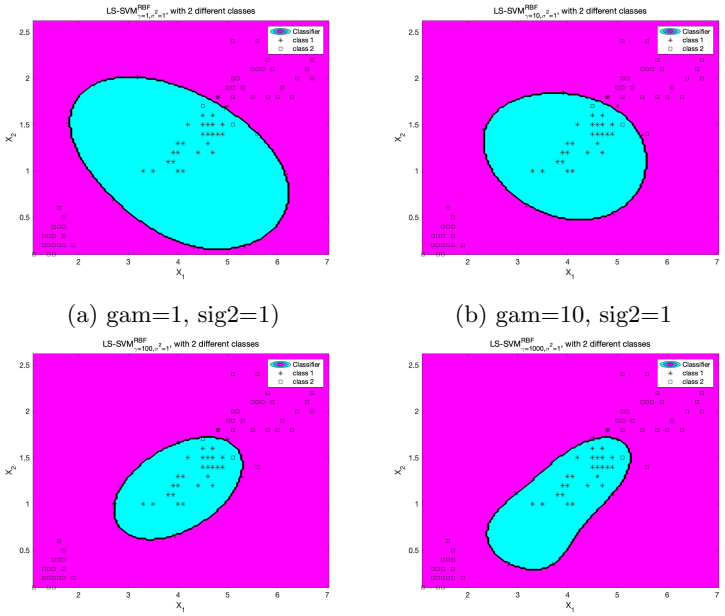


Figure 7: Polynomial Kernel: Degree 1 (error = 55%), Degree 2 (error = 5%), Degree 3 (error = 0), Degree 0 (error = 4)

As shown in the Figure 8, under the condition of fixing sig2, an experiment was carried out in which the value of gam was changed and degree was set to 9.



(a) gam=1, sig2=1) (b) gam=10, sig2=1

(c) gam=100, sig2=1 (d) gam=1000, sig2=1

Figure 8: The different choices of gam with fixed gam (sig2=1)

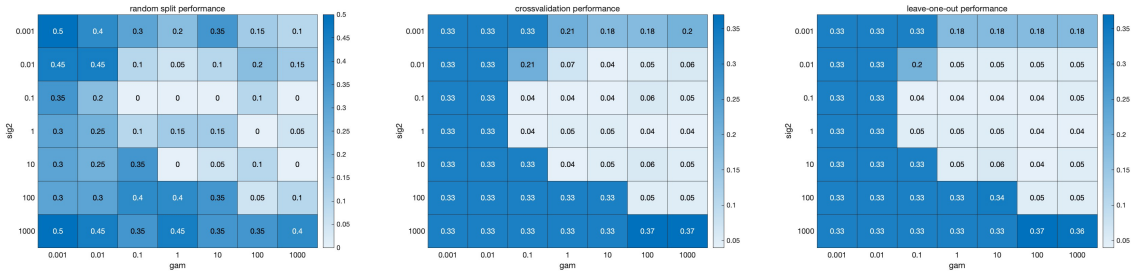
Table 1 shows the results of the experiment, with each row corresponding to a different gamma value. We assessed the performance of each experiment by measuring the error from the target value and recorded it in the table. These error values provide information about the effect of different gamma values on the experimental results.

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

gam	Sig2	error
0.001	1	50.00
0.01	1	50.00
0.1	1	0.00
1	1	0.00
10	1	0.00
100	1	0.00
1000	1	0.00

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.

In both cross-validation and leave-one-out validation, as the gam parameter decreases, the performance of the model improves. However, when comparing with random split validation, the random split method demonstrates better performance when the gam parameter is kept as small as possible, particularly in cases where the sig parameter is extreme, as shown in Figure 9.



(a) Random Split Performance (b) Crossvalidation Performance (c) Leave-one-out Performance

Figure 9: Performance

The performance of two automatic parameter tuning algorithms was tested and compared in terms of runtime, cost, and accuracy. The experiments were conducted with four runs for each algorithm. Table 2 shows that the optimal values for gam and sig2 varied significantly between runs. However, the cost values were relatively close across different runs.

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

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

In terms of runtime, the gridsearch algorithm took approximately twice as long as the simplex algorithm. This is because gridsearch performs an exhaustive search within a limited range, which makes it slower compared to the simplex algorithm. According to the user guide, the parameter tuning process consists of two steps. In the first step, the coupled simulated annealing (CSA) algorithm is used to select suitable initial parameter values.

In the second step, fine-tuning is performed based on the selected initial parameters using either the simplex or gridsearch algorithm. It is important to note that hyperparameter selection is a non-convex problem, which means there can be multiple local minima and no guarantee of finding the global minimum. Although CSA is a global optimization method, it cannot guarantee finding the global minimum. As a result, the optimized parameter values obtained from different runs can vary significantly.

In conclusion, the experiments demonstrated the variability of optimal parameter values and the impact on runtime and accuracy. The choice of parameter tuning algorithm can affect the efficiency and quality of the optimization process.

The primary goal of machine learning is to build models that can generalize well to unseen data. By evaluating the ROC curve on the test set, we can assess how well the model performs on new and unseen samples, providing a more realistic estimate of its performance in real-world scenarios.

Overfitting occurs when a model performs exceptionally well on the training data but fails to generalize to new data. By evaluating the ROC curve on the test set, we can detect if the model is overfitting by observing a significant drop in performance compared to the training set. This helps in identifying potential issues with model complexity and bias-variance trade-off.

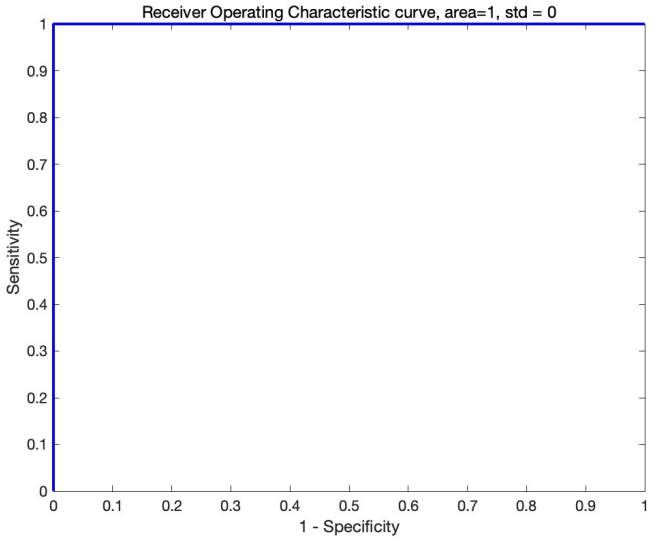
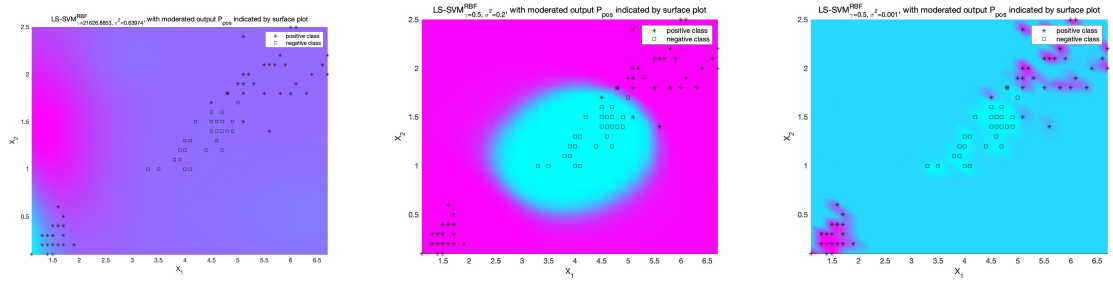


Figure 10: ROC Curve for Iris Dataset with Tuned Hyperparameters

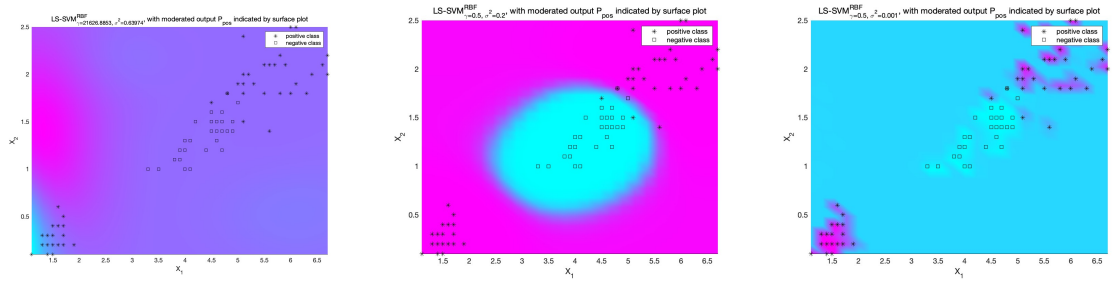
A higher AUC value indicates better discrimination power, where an AUC of 1 represents a perfect classifier, as shown in Figure 10. The hyperparameters obtained by using the coupled simulated annealing algorithm for optimization are gamma = 1.2529 and $\sigma^2 = 0.093912$. The minimized objective function value is 0.04.

The colors in the Figure 12 represent the probability that the data point belongs to the positive class, with blue representing low probability and red representing high probability. Larger values of gam and smaller values of sig2 lead to more confident predictions, with higher probability for positive points and lower probability for negative points. Conversely, smaller values of gam and larger values of sig2 result in lower confidence predictions and more overlap between the probability distributions of the two classes.



(a) gam=21626.8853,sig2=0.63974 (b) gam=0.5,sig2=0.2 (c) gam=0.5,sig2=0.001

Figure 11: Probability Estimates with Tuned Parameters

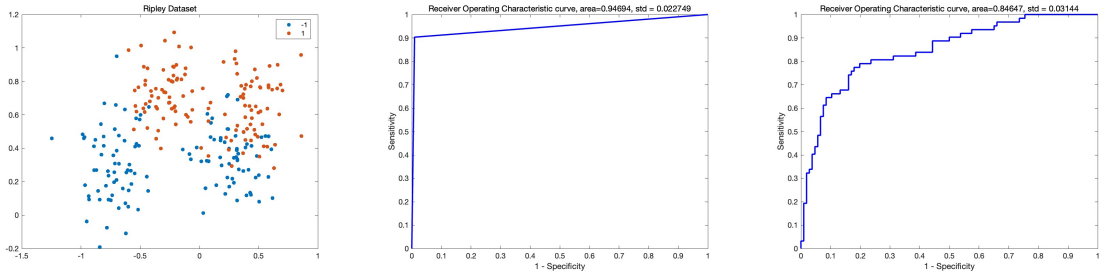


(a) gam=21626.8853,sig2=0.63974 (b) gam=0.5,sig2=0.2 (c) gam=0.5,sig2=0.001

Figure 12: Probability Estimates with Tuned Parameters

1.2 Homework problems

The Wisconsin Breast Cancer dataset looks clean, so we can start using a linear kernel. For the diabetes dataset, we can use the RBF kernel, but we need to adjust the parameters.



(a) Ripley (b) Wisconsin Breast Cancer (c) Diabetes

Figure 13: Probability Estimates with Tuned Parameters

Different models (linear, polynomial, and RBF kernel) were applied to the diabetes dataset with adjusted hyperparameters and kernel parameters. The data, which consists of 8 dimensions, was visualized by projecting it onto the first two principal components (PCs). However, the PCA plot revealed that the data points were not easily separable, indicating the need for more complex kernels.The misclassification errors for the polynomial, RBF kernel, and linear models were found to be 0.20833, 0.22619, and 0.23214, respectively. These relatively high error rates suggest that the models did not perform well on the dataset. Additionally, the ROC plot did not yield satisfactory results.

Given these observations, as shown in Figure 14, it is recommended to explore other methods that may offer better performance in improving the accuracy of the model on the diabetes dataset.

In the Wisconsin Breast Cancer dataset, the training set consists of 400 observations with 30 dimensions. To

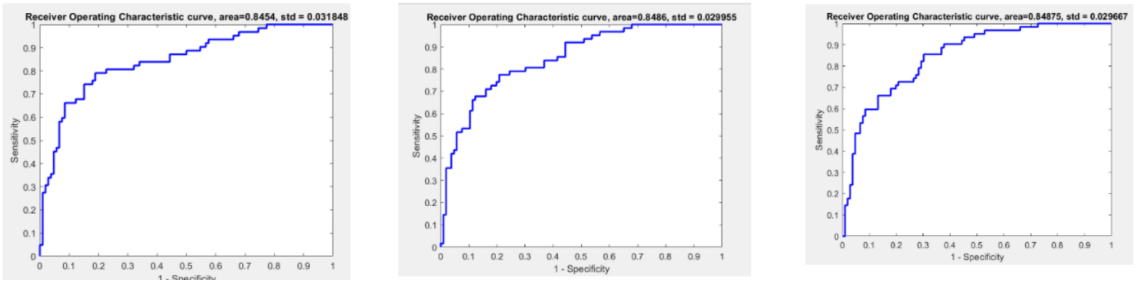


Figure 14: Diabetes Dataset

visualize the data, PCA is employed to project it onto the first two principal components. The Figure 15 demonstrates a clear separation between the two classes, indicating good separability. Three different kernel methods, namely linear, polynomial, and RBF, were applied to the dataset. The misclassification error rates for each method were computed as 0.0414, 0.1497, and 0.01775, respectively. The ROC plots for all three methods are displayed below, and they appear to be very similar. However, considering the lowest misclassification error and the favorable performance of the ROC plot, the RBF kernel is chosen as the preferred method. Its exceptionally low error rate contributes to a high level of satisfaction with this approach.

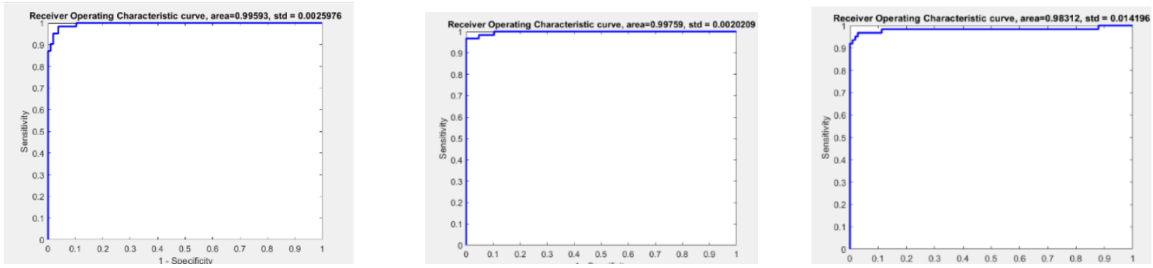


Figure 15: Wisconsin Breast Cancer Dataset

2 Function Estimation and Time Series Prediction

2.1 Exercises

2.1.1 Support vector machine for function estimation

The Figure 16 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.

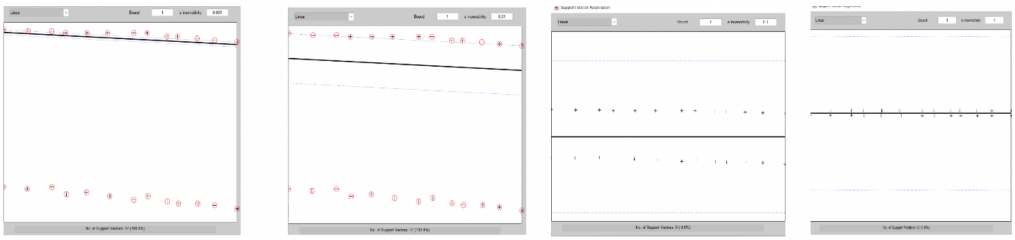


Figure 16: SVM Regression with Linear Kernel: Fixed Bound (1) and Varying e

The Figure 17 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.

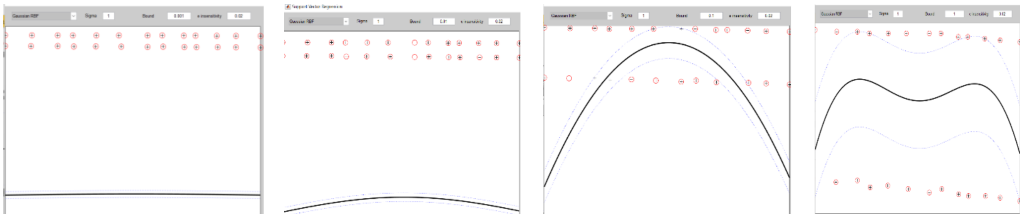


Figure 17: SVM regression with Gaussian RBF kernels for different boundary choices

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.

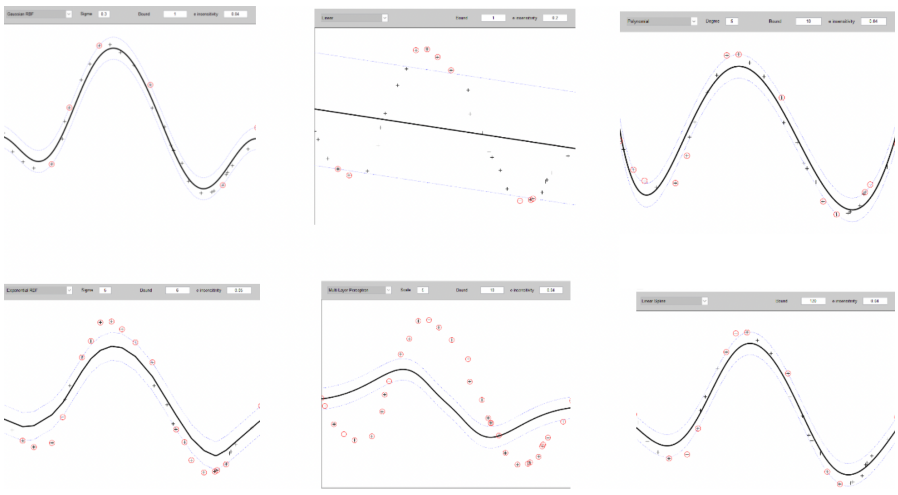
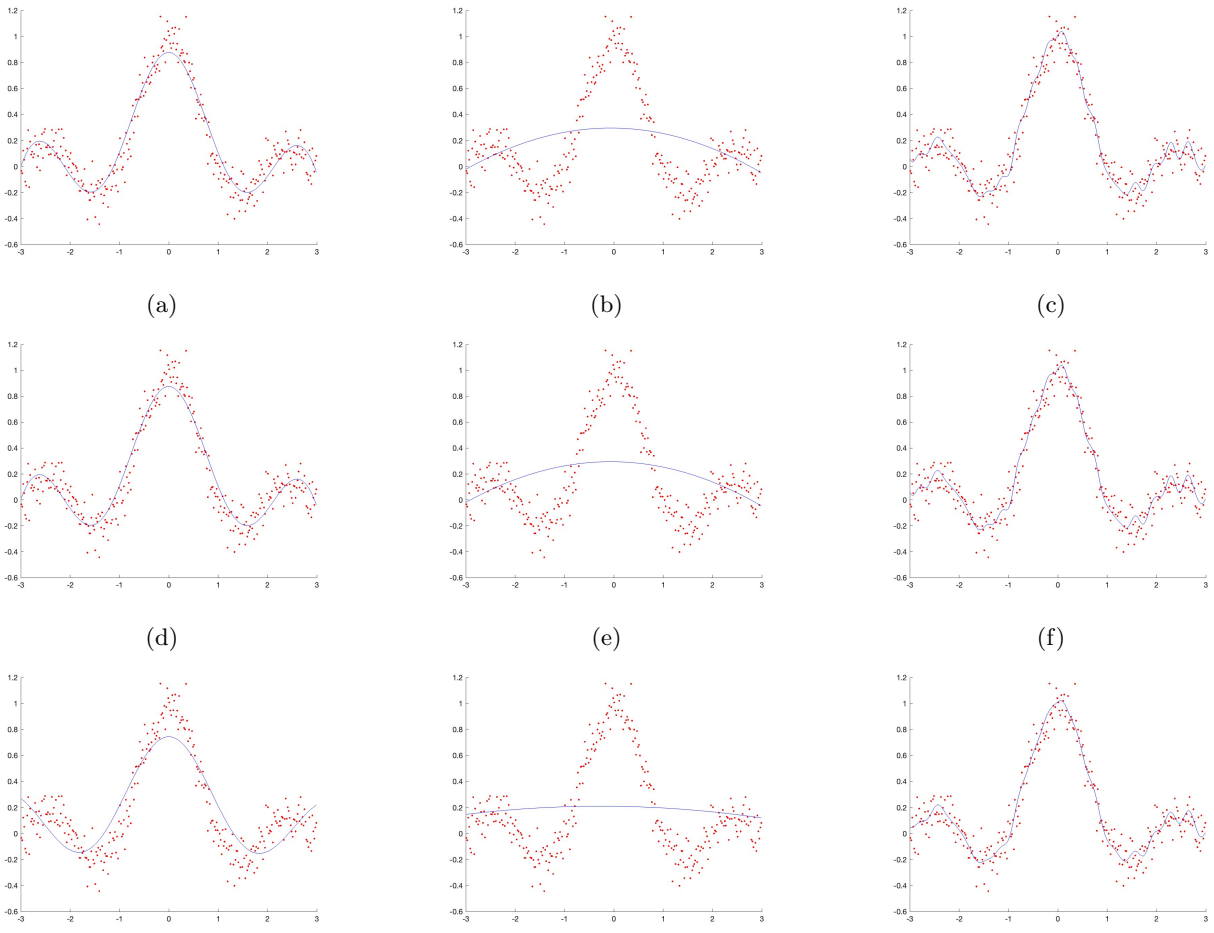


Figure 18: SVM regression with different kernels

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.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.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 19: Variation of Performance with gam and sig2

The mean square error (MSE) for each parameter combination (gam, sig2) is shown in the Table 3 below:

gam	sig2	MSE
10.000000	0.010000	0.010573
10.000000	1.000000	0.034955
10.000000	100.000000	0.132859
103.000000	0.010000	0.010879
103.000000	1.000000	0.015351
103.000000	100.000000	0.117996
106.000000	0.010000	0.010883
106.000000	1.000000	0.015234
106.000000	100.000000	0.117882

Table 3: MSE Values for Different Parameter Combinations

As the value of gam increases, the change of MSE is not monotonous. For example, MSE increases from 0.010573 to 0.015351 when gam is increased from 10 to 103, and then decreases slightly to 0.015234 when gam is increased to 106. This suggests that the optimal gam value may be different in different problems and datasets. As the value of sig2 increases, the change of MSE is not monotonous. For example, MSE increases from 0.010573 to 0.034955 when sig2 increases from 0.01 to 1, and then increases dramatically to 0.117882 when sig2 increases to 100. This suggests that the nature of the problem and the noise level of the dataset need to be considered when choosing the value of sig2. Different parameter combinations correspond to different MSE values. For example, when gam is 10 and sig2 is 0.01, the MSE is 0.010573, and when gam is 103 and sig2 is 100, the MSE is 0.117996. This shows that different parameter combinations have a significant impact on the performance of the model.

Within a given parameter range, different parameter combinations lead to different MSE values, and the choice of gam and sig2 has an important impact on model performance. In order to find the optimal parameter combination, we need to make further adjustments and optimizations according to the characteristics of specific problems and data sets. It is possible that there exists an optimal pair of hyperparameters, but it is not always guaranteed to be found. The choice of hyperparameters depends on the characteristics of the dataset and the needs of the application. Different datasets and problems may have different optimal choices for different combinations of hyperparameters. Therefore, finding a general optimal pair of hyperparameters can be difficult.

I used tunelssvm to adjust the gam and sig2 parameters. Since the results will change each time, the following Table 4 records the results of a certain time.

gam	sig2	MSE
Algorithm: simplex		
6.858020	0.182299	0.011039
7.268598	0.182912	0.011036
6.521872	0.178562	0.011045
6.089290	0.176178	0.011050
5.066028	0.175040	0.011062
7.393451	0.197301	0.011021
4.896748	0.169114	0.011070
5.283460	0.200386	0.011044
6.525511	0.169346	0.011057
Algorithm: gridsearch		
41.728163	0.405393	0.009634
11.703125	0.295961	0.009634
16.314841	0.292694	0.009626
16.834179	0.313789	0.009630
10.044849	0.267749	0.009628
33.977644	0.353664	0.009628
7.996208	0.265559	0.009635
11.654817	0.289715	0.009632
15.171408	0.280108	0.009624

Table 4: MSE Values for Different Parameter Combinations with Different Algorithms

Multiple runs using the tunelssvm procedure can help us observe changes in hyperparameters and their impact on the results. The simplex algorithm may lead to different optimal hyperparameter pairs, while the gridsearch algorithm is usually able to find the optimal hyperparameter pair within a given range of hyperparameters.

The model is initialized with suitable starting values, and in the first level of inference, the parameters alpha and b are optimized. In the second level, the regularization parameter gam is optimized. Finally, in the third level, the kernel parameter sig2 is optimized.This iterative process allows the model to refine and adjust its parameters based on the observed data and the posterior probabilities calculated at each level. By optimizing the parameters at each level, the model aims to improve its performance and fit the data more effectively. The iterative nature of the process allows for a more comprehensive exploration of the parameter space and helps in finding the optimal parameter values for the LS-SVM regressor.

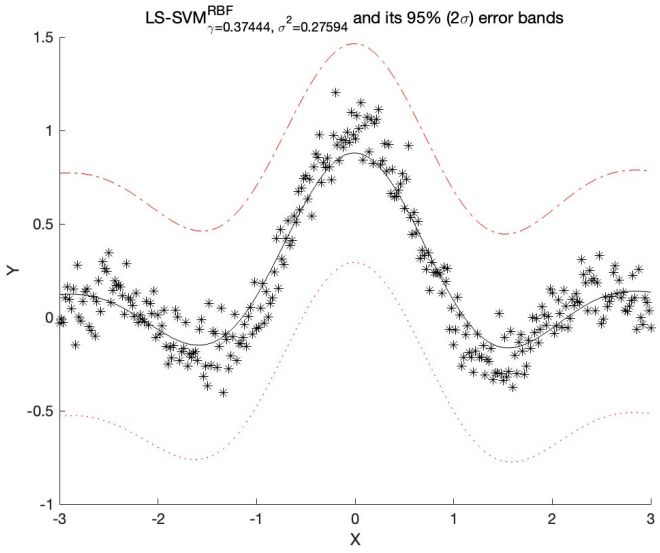


Figure 20: LS-SVM with 95% Error Bands

Starting from the initial parameter values (sig2 = 0.4 and gam = 10), the optimization process led to the determination of the final optimized parameters (gam = 0.37444 and sig2 = 0.27594). It is noteworthy that the Bayesian framework allows the estimation of 95% confidence intervals, which is a significant advantage, as shown in Figure 20.

2.1.2 Automatic Relevance Determination

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, as shown in Figure 21.The best fitting of the red data points (X1 vs Y) can be observed.

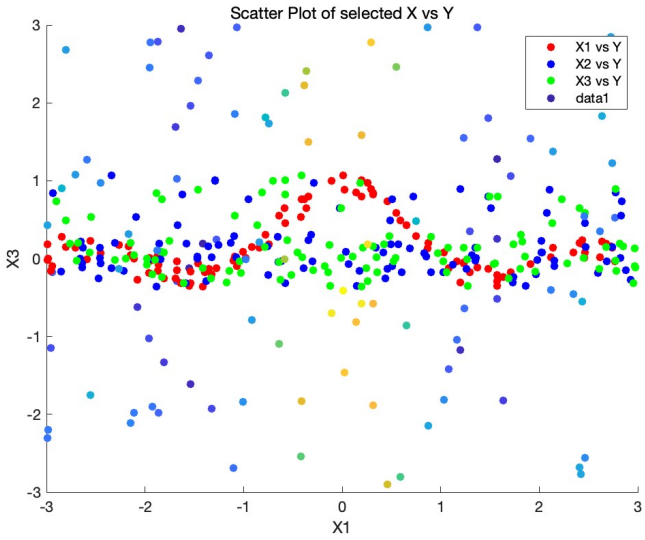
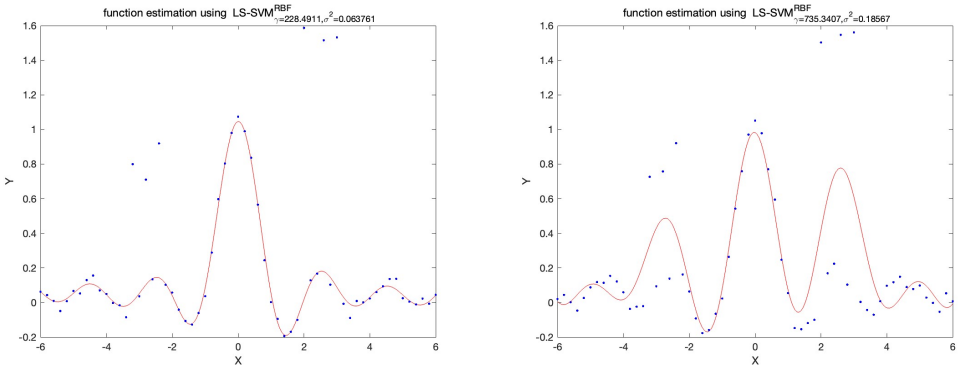


Figure 21: Automatic Relevance Determination with Backward Selection

As previously mentioned, the Bayesian framework is employed to select input dimensions in order to minimize the cost associated with the third level of Bayesian inference. Cross-validation is utilized to compare the performance of models utilizing different combinations of variables. The validation sets are used to calculate the error, which serves as a measure of the model's generalization performance. The value of sig2 is selected by minimizing the validation error.

2.1.3 Robust regression

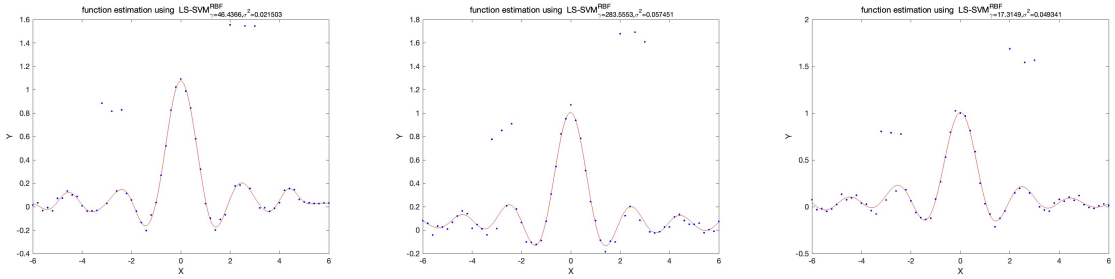
A comparison between the robust and non-robust versions reveals noticeable differences. The non-robust version is susceptible to the influence of outliers, leading to biased regression lines. In contrast, the robust version demonstrates improved performance by disregarding the outliers, resulting in more accurate regression estimates, as shown in Figure 22.



(a) Robust Regression (b) Regression Without Robustness

Figure 22: Comparison between the Robust and Non-robust

The fourth parameter of the tunelssvm function is set to 10, 'mae', where 10 represents the number of iterations for parameter optimization, and 'mae' indicates the usage of mean absolute error as the cost function. In addition, I modified the functions to whampel, whuber, and wlogistic in the robust regression case. Figure 23 is a comparison chart of these different weighting functions.



(a) whampel (b) whuber (c) maewlogistic

Figure 23: Comparison of Weighting Functions in Robust Regression

In non-robust regression, the regression line is highly influenced by outliers, resulting in biased regression results. However, robust regression with different weighting functions (whampel, whuber, wlogistic) helps mitigate the impact of outliers and provides more accurate regression estimates. By assessing the fit between the regression lines and the data points, a comparison of the different weighting functions can be made to determine the most effective one for the given dataset. Therefore, robust regression methods are more reliable when dealing with datasets containing numerous outliers, yielding robust and stable regression analysis outcomes.

For a given dataset and model configuration, different weighting functions will produce different results, as shown in Table 5. The results of the whuber weighting function performed best with the lowest mean squared error (MSE) and mean absolute error (MAE). The results of the whampel weighting function were next, which yielded slightly higher MSE and MAE values. The wlogistic weighting function gave the worst results, which yielded slightly higher MSE and higher MAE values. I think the whuber weighting function is a more appropriate choice under this data set and model configuration, which can provide better fitting effect and prediction accuracy.

Table 5: Comparison of Different Weighting Functions1.4

Weighting Function	MSE	MAE
whuber	0.1428	0.1312
whampel	0.1512	0.1320
wlogistic	0.1406	0.1336

2.2 Homework problems

2.2.1 Logmap dataset

Parameter search ranges, namely gamRange (1 to 10), sig2Range (1 to 10), and orderRange (5 to 15), were defined to encompass a wide range of values for each parameter. These ranges allowed for the exploration of various combinations of parameter values.By employing cross-validation, the best combination of parameters was determined to be gam = 10, sig2 = 5, and order = 5. This selection was based on the model's performance on the validation sets, where the goal was to minimize the validation error.

Figure 24 visually presents the results of this parameter selection process, showcasing the performance of the model with the chosen parameter combination.

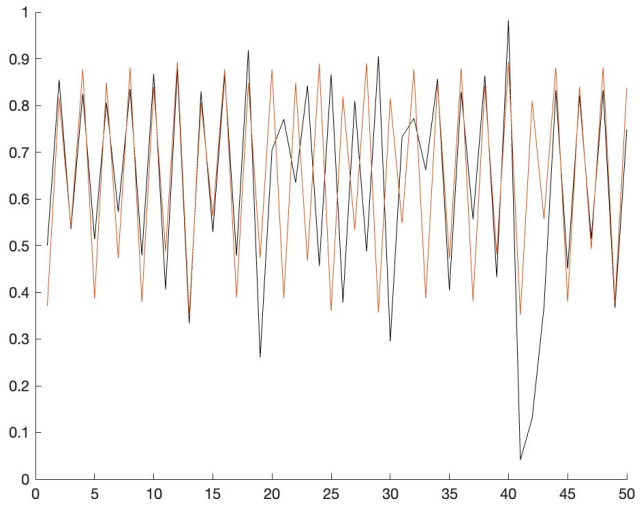


Figure 24: Parameter Selection Performance

The visualization of the prediction can provide insights into the best fits for various order values. However, it is important to acknowledge that the time series prediction based on the optimized parameter may not yield the desired or optimal results.

2.2.2 Santa Fe dataset

When the selected order is 50, it needs to be evaluated according to the specific situation. A higher order value can capture longer-term time dependencies, but may also increase the complexity of the model and the risk of overfitting. Therefore, whether it is appropriate to choose an order of 50 depends on the characteristics of the data set and the requirements of the problem.

It makes sense to use validation set performance to optimize hyperparameters and model order. Overfitting to the test set data during model selection can be avoided by using the validation set data for performance evaluation during the cross-validation process. By evaluating the performance of different hyperparameters and model orders, those parameter combinations that perform best on the validation set can be selected, thereby improving the generalization ability of the model. Optimizing hyperparameters and model order using validation set performance can also help tune model complexity. By experimenting with different hyperparameters and model orders, it is possible to find the optimal combination of parameters that allows the model to capture important features of the time series data without overfitting the training data.

After tuning the parameters (order, gam and sig2) and performing time series forecasting, we got the best parameter combination as gam=9, sig2=2, order=7. When forecasting with this combination of parameters, we observe that in the range of times less than 70, the forecasting works better. To visualize the results, we plot the original time series against the predicted results in Figure 25. According to observations, the

prediction results can accurately reflect the trends and changes of actual observations at some time points. However, when the time exceeds 70, we may notice some discrepancies between the predicted results and the actual observations.

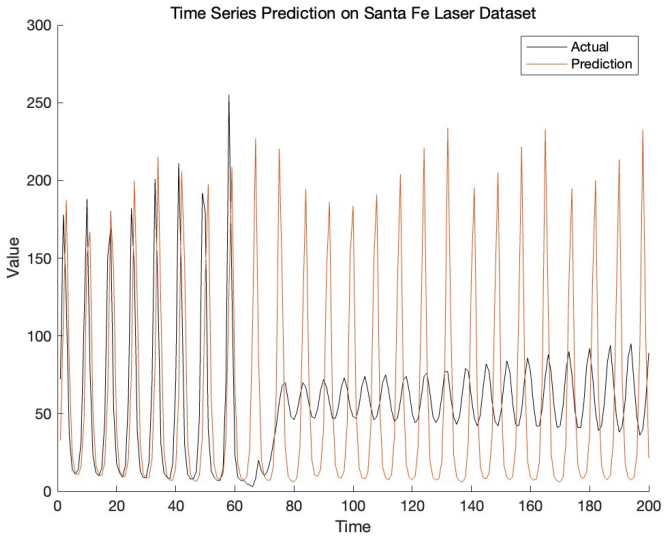


Figure 25: Comparison of original time series with forecasted results

3 Unsupervised Learning and Large Scale Problems

3.1 Exercises

3.1.1 Kernel principal component analysis

PCA works by reducing the dimensionality of the data and filtering out noise in the process. First, we construct the sine and cosine function data with Gaussian noise. Then, the PCA method is applied to extract the principal components, and the denoising results can be affected by the number of retained principal components. Next, we reconstruct the data using fewer principal components so that the reconstructed data has less noise compared to the original data. To quantify the quality of the denoising effect, we compute the reconstruction error, which measures the difference between the original and reconstructed data. Finally, we repeat the denoising process for different parameter values, such as the number of principal components and noise variance, and output the reconstruction error for different parameter combinations.

If the number of principal components (nc) is increased, a more accurate representation of the raw data will be obtained. However, increasing the number of principal components may also lead to overfitting and capture noise.

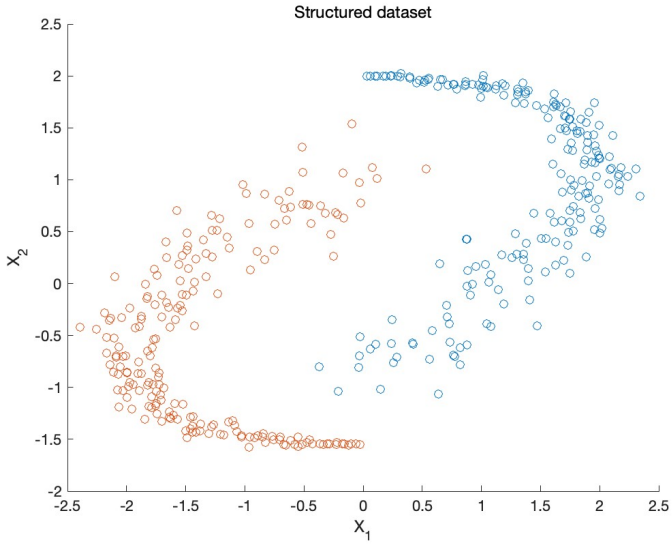
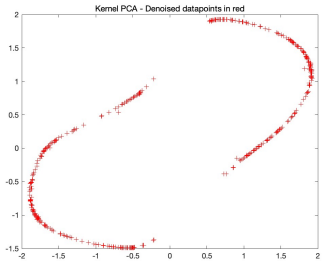


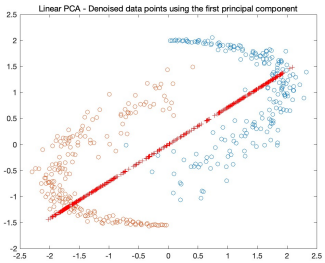
Figure 26: Original Situation

Linear PCA is limited to capturing linear relationships in the data, while kernel PCA can capture non-linear relationships. Linear PCA calculates the principal components directly from the data's covariance matrix, while kernel PCA requires computing the kernel matrix.

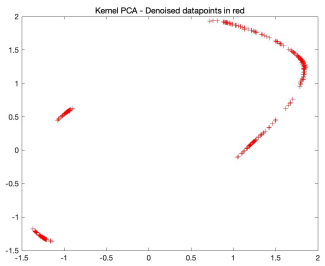
The number of principal components obtained is equal to the dimensionality of the data. The actual number of principal components retained can be controlled by the parameter nc. By changing the value of nc, different numbers of principal components can be obtained, and the corresponding denoising performance can be observed.



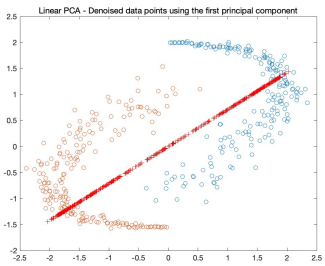
(a) robert(Lanczos)



(b) norobert(Lanczos)

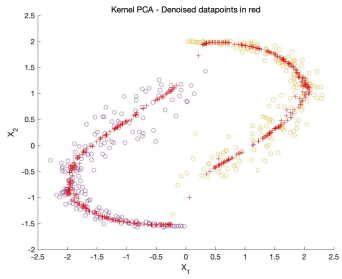


(c) robert(Nystrom)

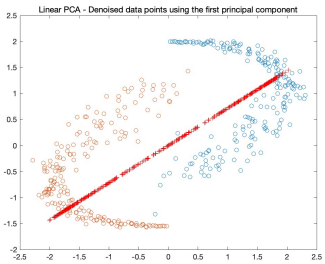


(d) norobert(Nystrom)

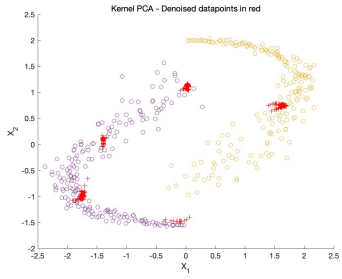
Figure 27: Comparison of Lanczos and Nystrom Methods(nc=6)



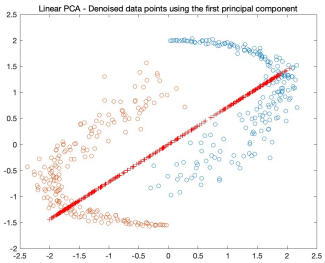
(a) robert(Lanczos)



(b) norobert(Lanczos)



(c) robert(Nystrom)



(d) norobert(Nystrom)

Figure 28: Comparison of Lanczos and Nystrom Methods(nc=10)

The analysis reveals that the lowest reconstruction error occurs when nc is set to 24 and sig2 is set to 10. In general, an ideal range for nc falls between 24 and 50, while an optimal range for sig2 is around 1. These parameter combinations demonstrate improved reconstruction performance within the given context, as shown in Table 6.

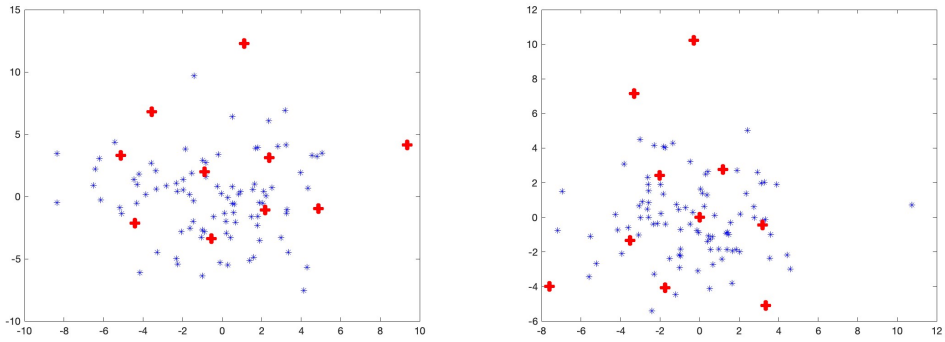
Table 6: Reconstruction Errors for sig2 and Number of PCs Combinations

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

3.1.2 Fixed-size LS-SVM

The number of features (dimensions) is smaller compared to the number of data points.The primal formulation of the model is computationally more efficient than the dual formulation.The primal formulation allows for easier interpretation and understanding of the model's parameters.

In Figure 29, blue markers represent all data points in the original dataset, red markers represent a fixed-size subset selected by the algorithm at each iteration, and blue markers are used to plot all data points in the original dataset. There are 100 data points in total, and each data point has two features.

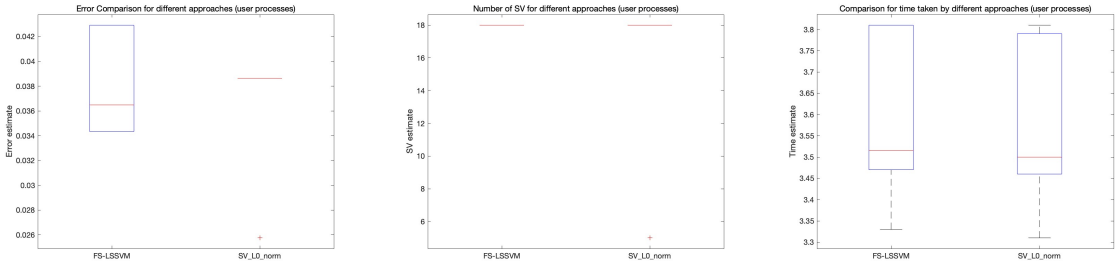


(a) sig=1 (b) sig=0.1

Figure 29: Subset Selection and Iteration Process

The effect of varying sig2 values on subset selection is illustrated in this analysis. When using a higher sig2 value, the selected subset points in the feature space are more widely dispersed. These selected points effectively represent the characteristics of each region. Consequently, the subset algorithms aim to converge towards a selection of data points that can effectively cover the entire feature space. By selecting these representative data points, the algorithms can construct a robust model that captures the essential features of the data distribution.

In Figure 30, when comparing fixed-size LS-SVM to l0-approximation with SV LO norm, both approaches achieved similar performance in terms of test errors and the number of support vectors used. However, there was a noticeable difference in computational time, with l0-approximation showing a slight improvement in terms of time efficiency.



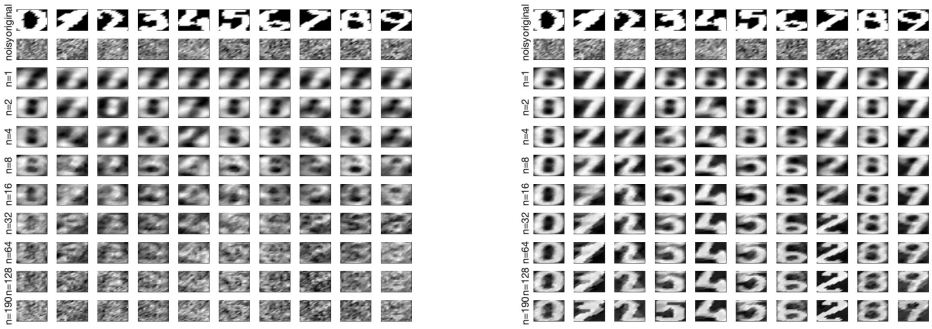
(a) Error Comparison for different approaches (b) Number of SV for different approaches (c) Comparison for time taken by different approaches

Figure 30: Comparison of LS-SVM and l0-Approximation

3.2 Homework problems

3.2.1 Kernel principal component analysis

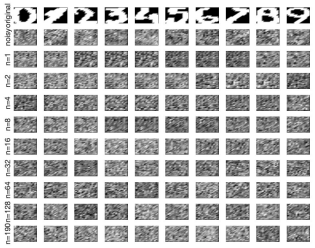
The Figure 31 shows the digital handwriting image denoising with linear PCA and kernel PCA with RBF kernel. The noise factor is selected as 1. We can see the numbers with a lot of noise from the second row of the figures, the numbers are impossible to be recognized. When the extracted components of kernel 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.



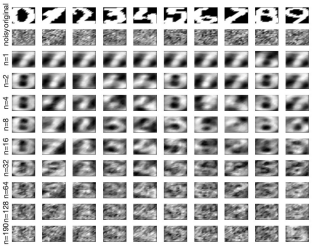
(a) linearpca (b) kernelpca

Figure 31: Denoising Handwriting with Linear and Kernel PCA

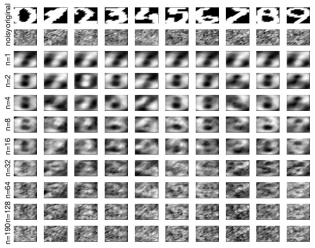
The Figure 32 shows the digital handwriting image denoising with linear PCA and kernel PCA with RBF kernel. The noise factor is selected as 1. We can see the numbers with a lot of noise from the second row of the figures, the numbers are impossible to be recognized. When the extracted components of kernel 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.



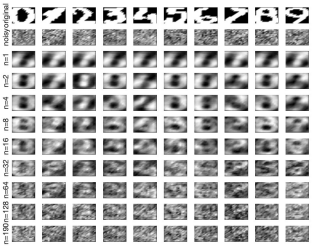
(a) kernelpca(sig2=0.0001)



(b) kernelpca(sig2=1000)



(c) linearpca(sig2=0.0001)



(d) linearpca(sig2=1000)

Figure 32: Comparison of sig2=0.0001 and sig2=1000

Sigmafactor	**sig2**
0.01	0.51297
0.03	1.4274
0.1	3.9717
0.3	11.0516
1	30.7517
3	85.5684
10	238.0991
30	662.525
100	1843.5151
300	5129.6903

Table 7: Denoising with Different sig2 Values

The generated Figure 33 shows the reconstruction error curve under different values of sigmafactor, as shown in Table 7. The figure contains three curves, representing the trend of the reconstruction error of the training set, validation set 1, and validation set 2 as a function of sigmafactor. We can determine that the sigmafactor value corresponding to the point where the verification set reconstruction error is minimized is the optimal parameter setting we are looking for. We found that the optimal sigmafactor value is 0.21544, and when applying this optimal value to the denoising operation, we get the optimal sig2 value of 11.0516.

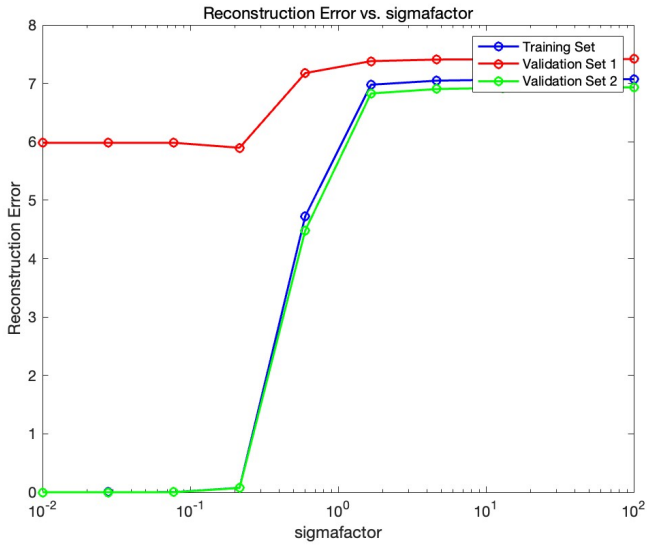


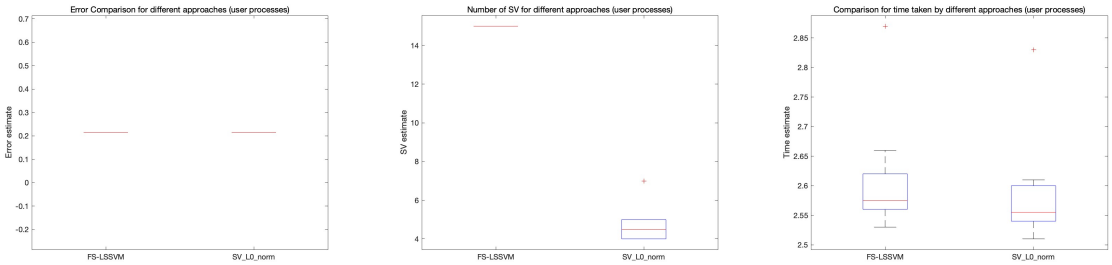
Figure 33: Reconstruction Error Curve for Different Values of Sigmafactor

With optimal parameter settings used, the reconstruction error can be further evaluated. According to the results, the reconstruction error of the training set is 0.075641, the reconstruction error of the validation set 1 is 5.8975, and the reconstruction error of the validation set 2 is 0.072685. These results show that under the optimal parameter setting, we can effectively reduce the reconstruction error, leading to more accurate denoising and reconstruction results.

3.2.2 Fixed-size LS-SVM

The Shuttle dataset was loaded, the first 700 data points were selected, and the labels were binarized. Except for the time attribute, the remaining 8 attributes are numeric attributes. There are 7 categories and each data point is assigned a category label.

According to the dataset information, about 80% of the data points belong to category 1. Therefore, based on the default classification accuracy, we can expect to achieve about 80% accuracy. However, based on the information provided, the goal here is to get 99-99.9% accuracy. This means that on this dataset, the performance goal of the classification algorithm is very high accuracy.

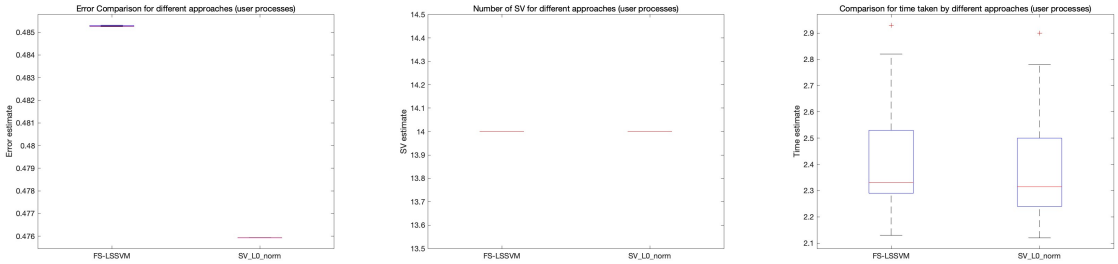


(a) Error Comparison for different approaches (b) Number of SV for different approaches (c) Comparison for time taken by different approaches

Figure 34: Shuttle dataset

FS-LSSVM and SV LO norm perform similarly in terms of error performance, number of support vectors and computation time, but FS-LSSVM may prefer to model data with more support vectors, as shown in Figure 34.

The S&P Letters dataset contains 20,640 data points with 9 attributes. The S&P Letters dataset is probably a regression problem rather than a classification problem.



(a) Error Comparison for different approaches

(b) Number of SV for different approaches

(c) Comparison for time taken by different approaches

Figure 35: S&P Letters dataset

The FS-LSSVM and SV LO Norm methods perform similarly in some respects based on comparisons of test error, number of support vectors, and computation time, but there may be some differences in computation time, as shown in Figure 35.