KU LEUVEN

Master of Artificial Intelligence

Project Report

Support Vector Machines Course Report

Nupur Jhankar r0766694

Prof. Johan Suykens
Artificial intelligence
KU Leuven

Contents

1 Exercise Session 1 : Classification

1.1 A simple example: two Gaussians

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

This exercise is to get an intuitive insight into what classification is. This sections has two simulated datasets. These datasets are created using MatLab function randn(). One of the dataset is centered around (1,1) and the other (-1,1).

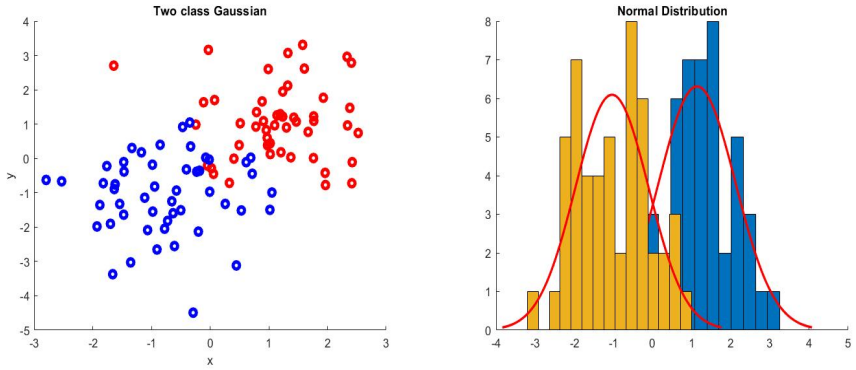


Figure 1: Simulated Datasets and Gaussian distribution

SVM mainly aims at minimizing the number of misclassification. This is done by widening the soft margin and the mid point would help in drawing the hyperplane. Support vectors are the data points close to class seperating hyperplane . We classify the above-simulated dataset we first plot them. we find the center point between two classes and draw a hyperplane. The function defining the hyperplane is $y = beta * x + bias$ This method is optimal because there is no significant variance in the data points and we don't have outliers. Therefore centers can be estimated with good accuracy. But the model is not perfect, as the classes are quite separable in this case. Classifier line is linear therefore the model cannot be used for non-separable cases and if some misclassification occurs.

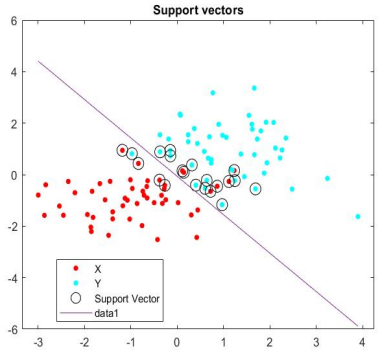
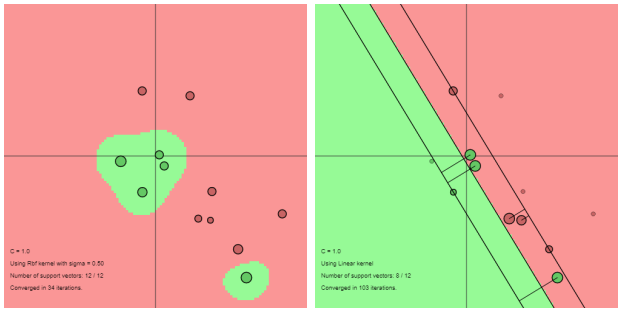


Figure 2: The simulated dataset with optimal classifier

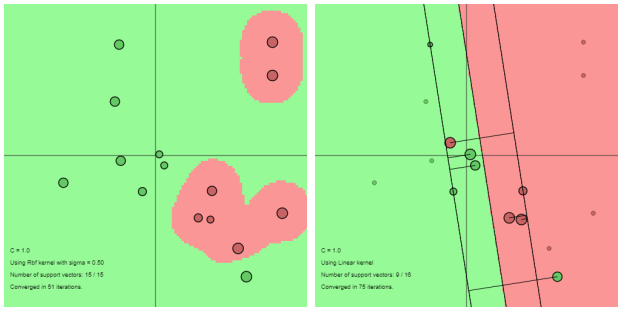
1.2 Support Vector Machine Classifier

In this section we experiment SVM classifer with a online application. This application allows us to add data points. We add data points to the graph and notice if the point is closer to the decision line, it plays bigger role for the model estimation. Also when data points is added to their own class (far from decision boundary), the impact on the model is not great.This is due to lagrangian multiplier a_k for every

data point. a_k should be greater or equal to zero.If a_k is a non zero value it contributes to definition of the decision boundary. These lagrangian values are called support values and the data point is called support vectors. If a_k corresponding to a data point is equal to zero it doesnt contribute to definition of decision boundary.Again this would mean that if we add data point in opposite class , that particular point would constitute within soft margin and would act as support vectors. When data points are added to opposite class it causes the decision boundary to shift.



As we can see that green data points form one cluster and red falls in global cluster. SVM classifies them and data points falling in soft margin contitutes of support vectors.



Now in the above images we see red data points forms the clusters and green data point forms global cluster. we also observe that the svm classifies the data points allowing misclassification of green point in red class.

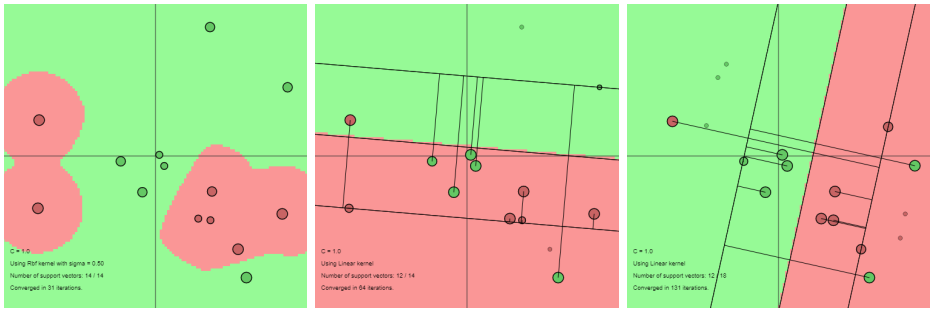


Figure 3: RBF and linear decision boundary

We also observe how addition of data point in respective class doesnot affect the classifier but when these data points are added to opposite class it vastly alters the hyperplane

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

The parameter C controls the slack of the SVM model. When the C is high tolerance for misclassification

is less and therefore a smaller margin. When C is larger we have higher tolerance for misclassification and hence a larger margin.

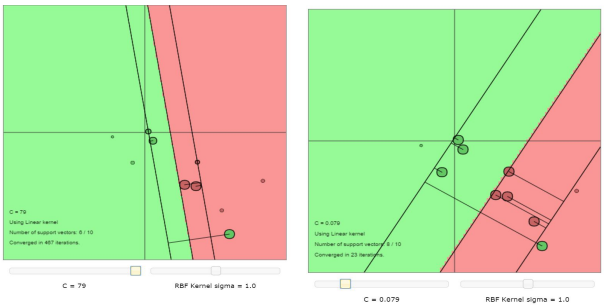


Figure 4: Role of Parameter C: large C and small C

The sigma defines the width of classification kernel. If sigma is small the radius of minority/underrepresented class is small.As we increse the sigma we obtain smooth patch seperation around the class.

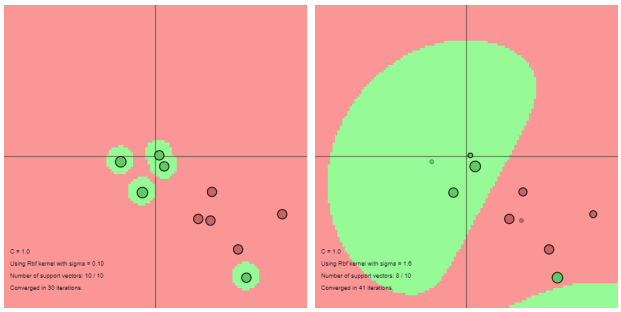


Figure 5: Role of Parameter sigma: sigma = .10 and sigma = 1.6

We have discussed the effect of parameter sigma and C in detail in next question

Difference between linear decision boundary and RBF kernel

When we use linear kernel instead of of RBF the misclassification steaply increses as this kernel can consider only linear decision boundary instead of complex pattern in data. When we add data points to RBF kernel overall seperation of the space is uneven. By adding a new red point the majority space become red and vise a versa. Hence we have some areas classified as red and others as green . The smaller space is in patches.Limitation of linear kerner is ,it is applicable only for linearly seperable data. this is overcome by RBF

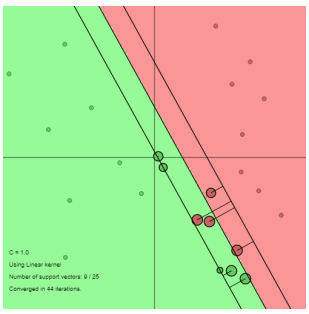
Question: What is a support vector? When does a particular datapoint become a support vector? When does the importance of the support vector change?

Support vectors are a subset of input data points that lie near to class determining hyperplane. They affect the position and orientation of this hyperplane. If we delete the support vectors position of the hyperplane will change. These are the points that help us in building our SVM. Support vectors have non zero Lagrangian multipliers a_k . After obtaining the dual representation using quadradic programming we apply Mercier theorem.

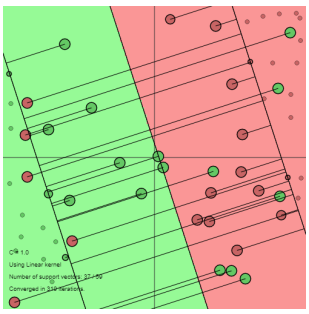
$$y(x) = sign\Big[\sum_{k=1}^{N} \alpha_k y_k K(x, x_k) + b\Big] \tag{1}$$

SVM has sparse property and not all data points are used to define decision boundary. Data point x becomes a support vector if the a_k is a non zero value and is near seperating hyper plane. Number of support vectors in SVM corresponds to number of hidden units in ANN. Importance of support vectors changes in seperable and non seperable cases.In seperable cases the nearest point to opposite class(near plane of seperation) are

support vectors.In this cases we dont need many support vectors to define the model.. The support vectors are the outliers of a particular class. In non seperable cases , Several data points are misclassified and here several support vectors are required t define the model.Below we can see in seperable case number of support vectors are far less than in non seperable case. Please note that bigger datapoints in black boundary is support vectors.

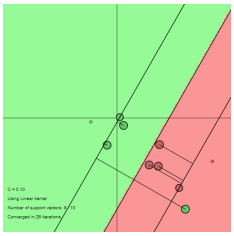


(a) Seperable Case

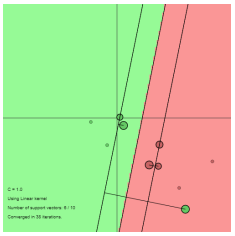


(b) Non Seperable Case

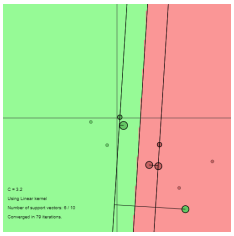
Question : What is the role of parameters C and sigma? What happens to the classification boundary if you change these parameters. Illustrate visually.What happens to the classification boundary when sigma is taken very large? Why?
'C' is Regularization term , that applies the trade-off between the training error and the flexibility of the solution i,e flatness of solution(allowed misclassification) of the model and minimization of error .If C is small it allows more misclassification.If C is the less ,then final training error will be less. But if we increase that value C to a very high value we might have the risk of overfitting and the generalization properties of the classifier will be compromised, because it will try to best fit all the training points. On the other hand if we reduce the value of C too much, our model will underfit the data, that is it will not learn the data properly. So its important to select the proper value of 'C'.



(a) C=0.1

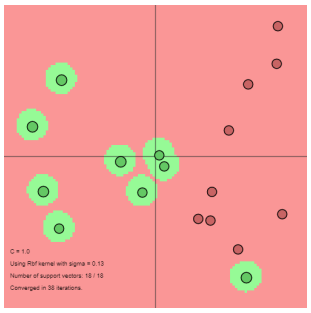


(b) C=1.0

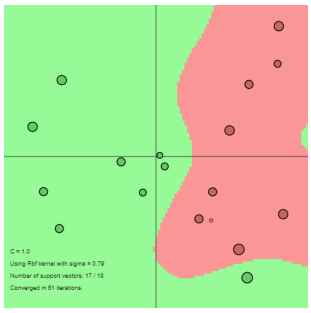


(c) C=3.2

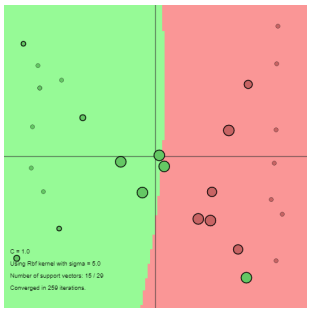
Sigma defines the width of RBF kernel.We observe that with a small sigma of 0.13 Radius around the underrepresented class is very small. We increse the sigma to 0.79 and observe a smooth patch formation around the class. when the sigma value is large say 5 the model becomes almost linear.Also the data point at the boundary of the class is of higher importance than that of the ones away from boundary



(a) sigma=0.13



(b) sigma=0.79

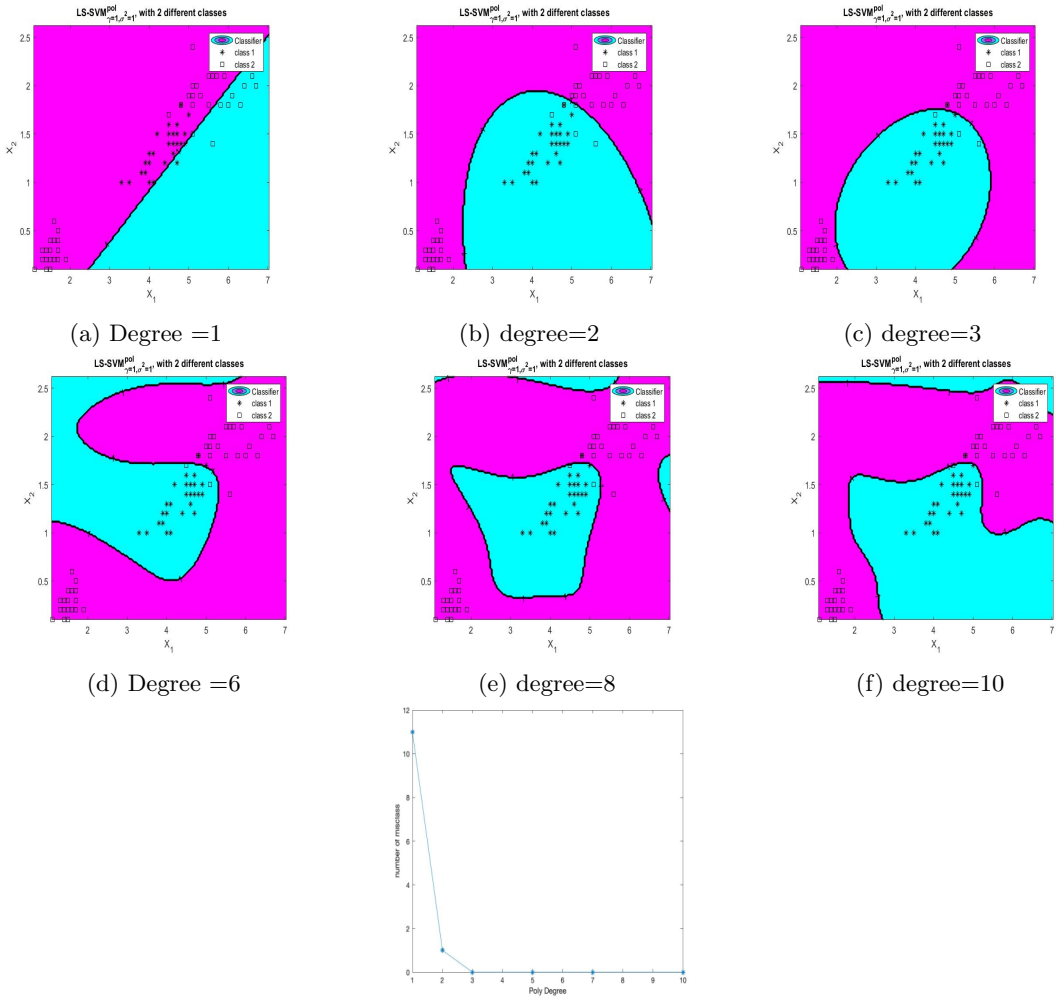


(c) sigma=5.0

1.3 Least-squares support vector machine classifier

1.3.1 Influence of hyperparameters and kernel parameters

Question : Try out a polynomial kernel with degree = 1, 2, 3, . . . and t = 1 (fix gam = 1). Assess the performance on the test set. What happens when you change the degree of the polynomial kernel?
The increse in degree of the polynomial results in less number of misclassification and error steaply reduces. Also For higher degree of polynomial ,error reduces to ZERO. The linear polynomial(degree=1) gives high error(55 percent) and 11 misclassification. For degree 2 there is a significant drop in error rate (5 percent) with only 1 misclassification. for degree 3 and higher there is no misclassification and error is zero. Although for degree 2 and 3 ,our classification curve is smooth but when the degree increses we get complex classification curve. For higher degree we have risk of overfitting so we need to have a balance so as to get a generalised model.



(a) Degree =1 (b) degree=2 (c) degree=3

(d) Degree =6 (e) degree=8 (f) degree=10

Figure 9: Variation between degree and misclassification

Question: Let's now focus on the RBF kernel with squared kernel bandwidth σ^2. Try out a good range of different sig2 values as kernel parameters. Assess the performance on the test set. What is good range of sig2? Fix a reasonable choice for the sig2 parameter and compare the performance using a range of gam. What is a good range of gam?
We fix gam as 1 and take σ^2 as [0.01, 0.1, 0.5, 1, 10, 25,50] .We get zero misclassification for values between 0.1 and 10. Error significantly increses to 50 percent fr higher values than 25.Performance wont be optimal

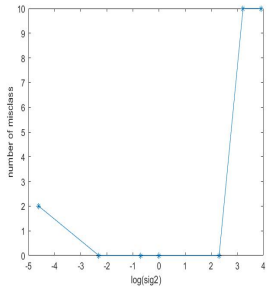
for non linearly seperable datasets. optimal chose for sig2 is [0.1,10]. To represent RBF kernel with two samples x and x', in some input space,

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

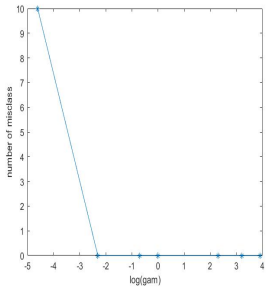
where x-x'is eucliclidian distance.

σ^2	**Misclassification**	**Error in percentage**
0.01	2	10
0.1	0	0
0.5	0	0
1	0	0
10	0	0
25	10	50
50	10	50

Table 1: error rate with different sigma for fixed gam =1



(a) number of misclass vs log(sig2) for fixed gam=1



(b) number of misclass vs log(gam) for fixed sig2=1

Next we fix sig2=1 and test with a range of gamma = [0.01, 0.1, 0.5, 1, 10, 25,50]. Here we observe that the classifier gives excellent results(zero error) with zero misclassification for gamma greater than or equal to 0.1.Very high value of gamma is not advisible since minimization of error will be compromised.Aceptable range = [0.1 , 1]

σ^2	**Misclassification**	**Error in percentage**
0.01	10	50
0.1	0	0
0.5	0	0
1	0	0
10	0	0
25	0	0
50	0	0

Table 2: error rate with different sigma for fixed gam =1

Comparison with SampleScript iris.m :
We observe that behaivioir of this script is almost same as before. For linear polynomial we get misclass = 11, error rate = 55.00 percent and for polynomial of degree 5 we get error of zero.This justifies the above

observation.Here also the fized gam is taken as 1 but varied sig2 is [0.01, 0.1, 1, 5, 10, 25]. But we observe a similar pattern as above. Now we change fixed gam to 2 and test for same range of sig2.We plot a graph between log(sig2) and num of misclass and observe that both exhibits similar behaivior.

1.3.2 Tuning parameters using validation

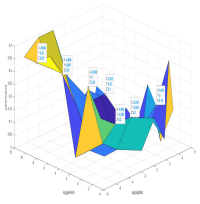
Question: Compute the performance for a range of gam and sig2 values. Use the random split method, 10-fold crossvalidation and leave-one-out validation. Visualize the results of each method: do you observe differences? Interpret the results: which values of gam and sig2 would you choose?

We split the iris dataset into training and validation set using 3 different automatic tuning algorithms viz random split, 10-fold cross validation and leave one out. We visualize the result using 3D mesh. We consider the following values of gamma and sig2:
gamlist=[0.1, 1, 10, 100, 1000, 10000, 100000]
sig2list=[0.001, 0.01, 0.1, 5, 25, 50, 100] We use random split to split 80 percent to training set and 20 percent to validation.We obtain following results :

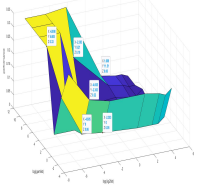
σ^2	**gamma**	**performance**
0.001	0.1	.2
0.01	1	0
0.1	10	0
5	10^2	0
25	10^3	0
50	10^4	0.1
100	10^5	0.35

Table 3: Performance while using random split tuning algorithm to split train and test set with given range to sigma2 and gamma

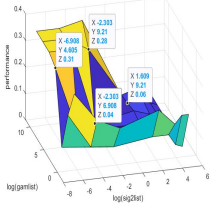
In K cross validation we split the trainingdata into K parts. Then train the data with K-1 parts and keep one part for validation.We repeat this step K time , and every time validation set will be different. This works well in case of less data and ensure good generalization. Leave-one out is similar to cross validation. Here we have N folds , N= number of data points. We perform the taining with N-1 data point and validate with last data point. we repeat this N time and average out the error.
We perform 10 fold cross validation where data is split into 10 groups and then one of the group is held as validation and rest as training. We retrain it covering every group as validation. This is useful when data is less. We reperform the cross validation by spliting the data into 5 groups and hence performing 5-crossvalidation. Then we perform leave one out cross validation on dataset. and visualize the results in 3D surface plots. The visualization is as follows:



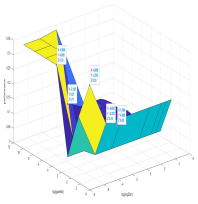
(a) Random slplit



(b) 10 -Cross validation



(c) 5-Cross validation



(d) leave one out

We notice that random split has a higher variation than both the cross-validation methods. The reason being for cross-validation methods we train the model on training parts and test on validation part, and perform this k times where validation group takes turns. The total error is calculated by taking an average of error at each iteration. That's the reason we have clean curves between the less performing combination

of gamma and sigma and high performing combination (yellow being high performing and violet being low performing)
From the figure and performance matrix we obtained we conclude that best combination of gamma and sig2 will be:
gamma = 10 , sig2 = 0.1
gamma = 100 , sig2 = 5,
gamma = 1000 , sig2 = 25.
Question: Why should one prefer cross-validation over simple validation (random split)? How to choose the value of k in k-fold cross-validation?
Cross-validation is preferred over simple validation due to various reasons:
1) When we have less data Cross-validation is preferred since it helps in better generalization since it uses up all the data to define the model. For Cross-validation we build K different models, so we make predictions on all of our data.
2) Error obtained from each model from K-fold is averaged out. This way variance of the result is minimized.
3) In case of a random split, the data set is randomly split and validation set occurs only once we might have the risk of bias since a good validation set is chosen by chance. and the visible results might not be representative of underlying distributions and might contain outliers. In Cross-Validation, we get K times more error metrics and helps us to draw an important conclusion both about our algorithm and our data.
We usually take the value of k as 5 or 10 although it is not an established rule. With a higher value of k, the difference between the size of the training set and subsets for resampling diminishes. When the reduction in this difference, the bias of the model decreases. There is a bias-variance trade-off associated with the choice of k in k-fold cross-validation technique. Generally, k-fold cross-validation is performed using k = 5 or k = 10, as these values don't give high bias or high variance.
The choice of K also depends on the size of the data set. For Smaller data, K should be as high as N(leave one out) since we need more data to train the model. For a large dataset, we go for 10 folds. Another option is to take 5 folds where reserve 20 percent data for validation at each step.

1.3.3 Automatic parameter tuning

Question: Try out the different 'algorithm'. What differences do you observe? Why do the obtained hyperparameters differ a lot in different runs? What about the cost? Computational speed? Explain the results
We run grid and simplex algorithm for 7 different iterations and achieve the following results:

Simplex			Grid search		
gamma	**sig2**	**cost**	**gamma**	**sig2**	**cost**
0.8173	0.16534	0.04	5.3	57.84	0.03
3.6804e+03	3.38	0.03	0.755	0.1126	0.04
0.067	0.702	0.04	1.141244	0.0274	0.04
1.2090e+04	3.78	0.04	2.5	0.0686	0.04
2.1311e+04	0.45	0.02	1.4486e+04	0.12	0.02
2.9490e+04	0.421	0.04	0.0988	1.5595	0.03
0.1715	2.11	0.04	1.4005e+04	1.7874e+03	0.03

Table 4: tuned pair of parameter

Training time 7 iterations for Simplex method 2.6 seconds Training time 7 iterations for grid search method 4.66 seconds.

Here a range of parameters are defined and then it executes K-fold validation. then the optimal pair is selected. Simplex and grid-search converge differently to minima. Grid search first evaluates grid over various parameters then select the minima. Simplex starts with random combination then converges to closest minima.

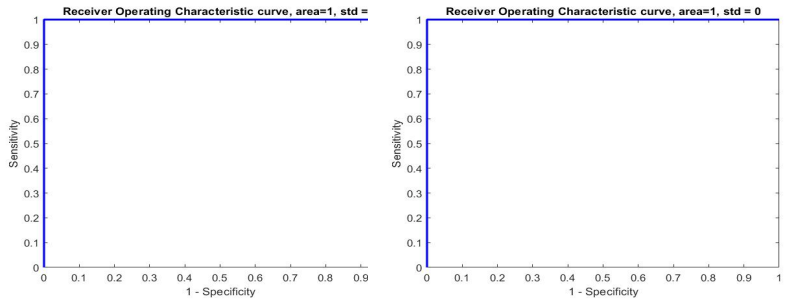
As observed, Grid search is slow as it spends time investigating hyper-parameter settings that are no not optimal. Nelder-Mead simplex algorithm is a better solution, which doesn't require the calculation of

gradient information and is easier to implement. Simplex is almost 2x times faster than grid search.

From the above result, we observe both simplex and grid search gives a similar cost (3 percent to 4 percent) for all combination of gamma and sig2 with few exceptions which gives 2 percent. Optimal gamma and sig2 combination can have a wide range so each iteration we obtain different combination as output.

1.3.4 Using ROC curves

Question: In practice, we compute the ROC curve on the test set, rather than on the training set. Why?
We create ROC curve by plotting a graph for True positive rate(TPR) against False positive rate(FPR). This curve helps in performance evaluation/generalization of the model on unseen data (like confusion matrix). Training fits the model to train data and performance will be optimized on train data. The major purpose of any model is to achieve generalization and produce good predictions on unseen data on unseen data. Hence any evaluation of the model is performed on unseen data and so ROC is computed on unseen data.
Generate the ROC curve for the iris.mat dataset (use tuned gam and sig2 values). Interpret the result
For this part we take tuned pair of gamma and sig2 that we get by running tunelssvm function for both simplex and grid search algorithms.For a certain tuned combination of gamma and sig2 we get perfect ROC curve. and that shows that model is perfect classifier as we can see in the figure below.



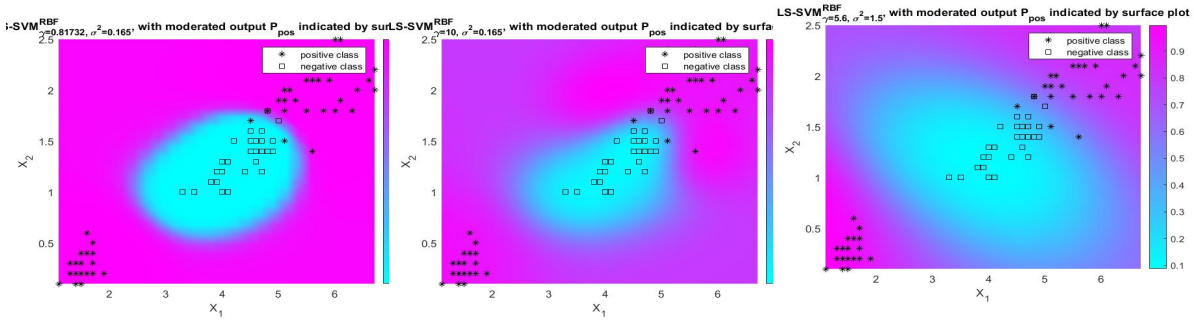
(a) simplex algorithm Gamma=4.43,sig2=11.73 (b) Gridsearch algorithm Gamma=2.5,sig2=0.0686

Figure 12: ROC Curves

1.3.5 Bayesian framework

Question :How do you interpret the colors of the plot?Change the values of gam and sig2. Visualize and discuss the influence of the parameters on the figure.
.In Bayesian framework ,the Support vector Machine is considered as a graphical model where parameters are connected through probability distributions . This extention allows the application of Bayesian techniques to SVMs, like flexible feature modeling, automatic hyperparameter tuning, and predictive uncertainty quantification. In the Bayesian inference there is no hard definition of decision boundary. Bayesian framework is modification over the svm classifier and a classifier that has output as probablity distribution. We can visualize this in figure 13.We dont have a hard decision boundary, instead we can see 2 color(class) viz pink and cyan merging each other resulting in voilet area (which is a merge of 2 classes).We can see in the colorbar. The top of the colorbar is pink,bottom part is cyan and middle(the part where the seperation occurs) is voilet.The data points in voilet region has more or less equal probablity of belonging to either class. (In svm terms -¿support vectors) We try different tuned combination of gamma and sig2. We first try with a tuned combination of gamma and sig 2 .With gamma around 0.8 and sig2 around 0.165 we observe the classification boundary is more concrete and very less area is voilet. In second figure the sig2 is unchanged but gamma is get to 10. here the outlier point is assigned voilet color(confused probablity). We notice that the voilet area/ confused increses and decision boundary is not concrete atall. In this combination considerable amount is voilet region is there with 50-50 probablity. when gamma is set to 0.69 and sig2 to 6.5 we get almost a

smooth classifier . Here the uncertainty of the model has been increased as and the cyan class has been enlarged compared to the tuned case.



(a) Gamma=0.81732,sig2=0.165 (b) Gamma=10,sig2=0.165 (c) Gamma=0.69,sig2=6.5 (with colorbar)

Figure 13: Bayesian inference grapf

1.4 Homework

Question : Visualize the data. Inspect the data structure: what seems to be important properties of the data? Which classification model do you think you need, based on the complexity of the data? Try out different models with tuned hyperparameter and kernel parameters. Compute the ROC curves. Which model performs best? Which model would you choose? Are you satisfied with the performance of your model? Would you advise another methodology?

1.4.1 The Ripley dataset

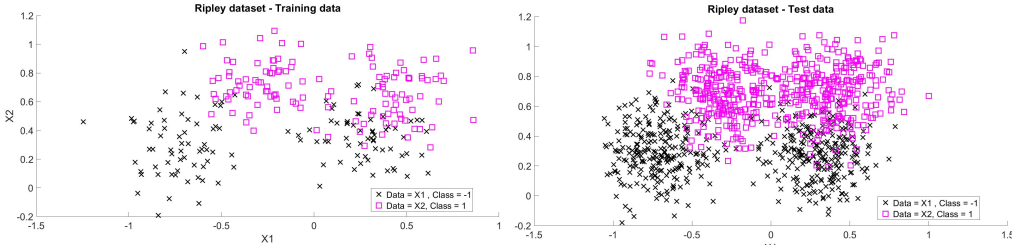


Figure 14: Training and test set of ripley dataset

Ripley dataset consist of 2 classes with 250 data points in trainset and 1000 in testset. We notice that 2 classes are not completely seperable so a non linear classifier might be optimal to such dataset.Next we use Automatic parameter tuning to obtain optimal parameter values for linear , polynomial and RBF kernel SVM function.

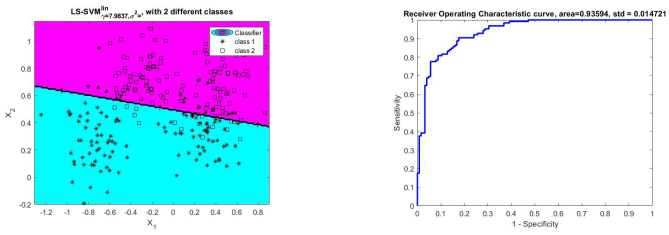


Figure 15: LSVM tuned with simplex method(Linear Kernel)

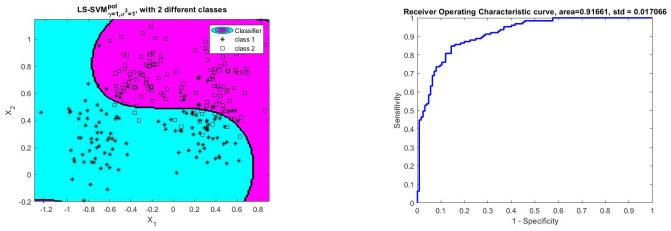


Figure 16: LSVM tuned with simplex method(polynomial Kernel , degree 3)

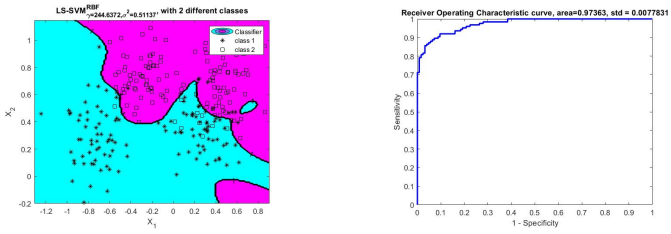


Figure 17: LSVM tuned with simplex method(RBF kernel)

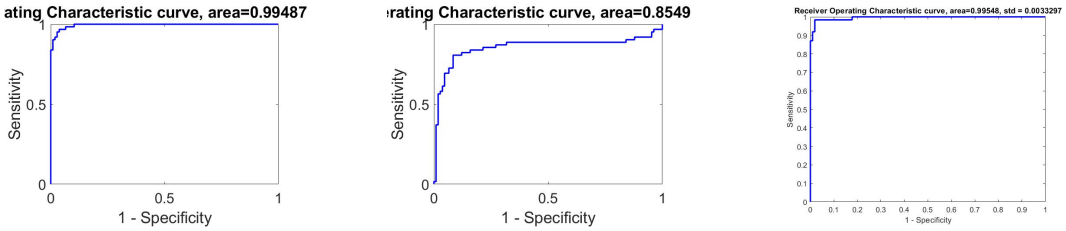
LInear kernel doesnot give a good performance because the dataset is not linearly seperable.The best ROC is achieved by RBF kernel.So ROC curve conclude that Rbf model is the best and gives smallest error on test set.

Linear kernel		Polynomial kernel			RBF kernel		
gamma	**error(%)**	**gamma**	**degree**	**error(%)**	**gamma**	**sig2**	**error(%)**
6.733	10.8	0.01	3	9.2	0.44138	0.061	10.6
0.0064777	10.4	0.25	4	9.9	0.34	0.044	10.6
0.043017	10.6	1.65	5	10.2	0.104	0.041	9.8
5.0027	10.8	0.65	7	9.7	0.057	0.35	10.1
0.0020614	10.8	0.85	10	12.6	13.2	0.18	12.6

From above table we conclude that out of all tested combination , best combination is Gamma= 0.65 and degree of polynomial =7 and this give lowest error of 9.7 with 97 misclassifications.We run the model in RBF kernel for 5 iteration with different combination of gamma and sig2. We get the lowest error of 9.8 percent with 98 misclassification with gamma= 0.104 and sig2 = 0.041.

1.4.2 Breast Cancer Dataset

No of Data sample 569, Training Set 400 points , Testing Set 169 points. Num of Input Variables 30 ,Diagnosis M= malignant, B = benign The dataset is higher-dimensional therefore we cannot visualize properly it in a 2D or 3D graph. Due to the complexity of the dataset, the Polynomial and RBF kernels will be Prefered. The ROC curves shows that RBF performs better than Polynomial kernel .From the table and ROC curve we conclude that RBF kernel gives the best performance with least error. the optimal pair or sig2 and gamma is generated and we calculate the misclassification on that.We observe that the data is multi dimentional with multiple features hence linear kernel is not advisable but we still run our tests on that.



(a) Linear kernel (b) Polynomal kernel (c) RBF kernel

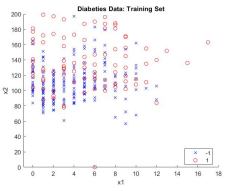
Figure 18: ROC curves with various kernel

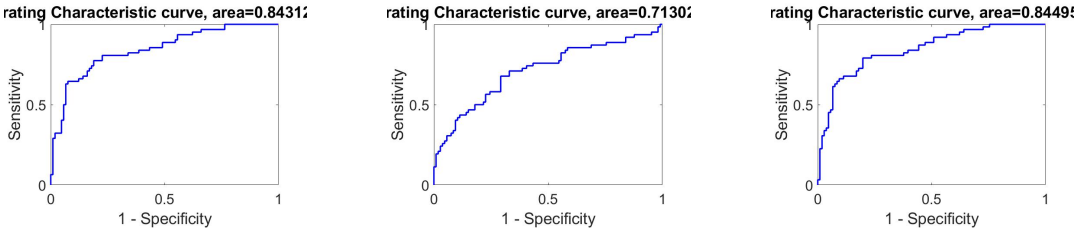
We display the roc curve of linear and polynomial kernel. We observe that although both perform equally good RBF gives a better performance.

Linear kernel		Polynomial kernel				RBF kernel		
gamma	**error(%)**	**gamma**	**sig2**	**degree**	**error(%)**	**gamma**	**sig2**	**error(%)**
1.1202	4.7337	0.0004498	13.669	5	13.018	32.301	23.488	2.3669
0.17131	3.5503.4	1455.5	253.68	5	3.5503	10.497	43.528	2.3669
22.799	4.7337	9267.2	76.406	3	3.5503	34.243	42.987	2.3669

1.4.3 Diabeties dataset

No of Data Points 600, Train Set 300 points, Test Set 169 points, No of Input Variables 8. Dataset is extreamly difficult to classify due to mixed complications.Analysing the ROC Curves and also the Errors generated by the models, we conclude that the RBF Kernel performs better than the polynomial Kernel. Moreover, when applying parameters tuning for polynomial kernel, the generated gamma value, degree and error generated were almost same over multiple iterations.





(a) ROC for Linear kernel (b) ROC for polynomial kernel (c) ROC for Rbf kernel

Linear kernel		Polynomial kernel				RBF kernel		
gamma	**error(%)**	**gamma**	**sig2**	**degree**	**error(%)**	**gamma**	**sig2**	**error(%)**
4.4257	21.429	0.0004498	1.44	5	27.018	34.668	1409.6	22.024
0.01157	24.405	0.0077852	1.84	5	29.5503	887.8	637.2	22.024
0.023042	22.619	1.89812	15.23	3	35.5503	66.074	242.94	21.4
0.014728	23.81	23847	2.88	5	36.905	20.174	791.2	21.4
3.7151	21.429	0.028079	26.99	3	29.167	21.6	501.04	20.833

From the Roc curve and the table above its safe to conclude that RBF kernel gives better performance than polynomial and linear kernel.Data is not linearly seperable andis multi dimentional so linearl kernel is not advisable to use. The best combination of gamma and sig2 is 21.6 and 501.04 that yeilds lowest error.

2 Exercise Session 2: Function Estimation and Time Series Prediction

2.1 Exercises

2.1.1 Support Vector Machine for Function Estimation

Question: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?

We create a dataset that lies inline and is linearly regressible. We the uiregress function to answer this question.In e-insensitive loss function, the epsilon is the error in function estimation (can also be considered as allowed misclassification by the model during classification/regression) .e-sensitivity refers to the tolerence factor of the model. This means that if the error is smaller / equal to epsilon,e, then the error is considered as 0, otherwise it is error - e.We can say that missclassification is tolerated till e. To examine the infuence of e we keep bound constant(0.1) and we modify the values of e starting from value e = 0.01. This would mean that we are allowing minimum misclassification in the model. The regression model performs well and seems fits the data . As the e value increases ,performance of the Regression model worsens, since the tolerance threshold of error is high. When e is set to 0.75, then the obtained model doesnot fit the data points at all.

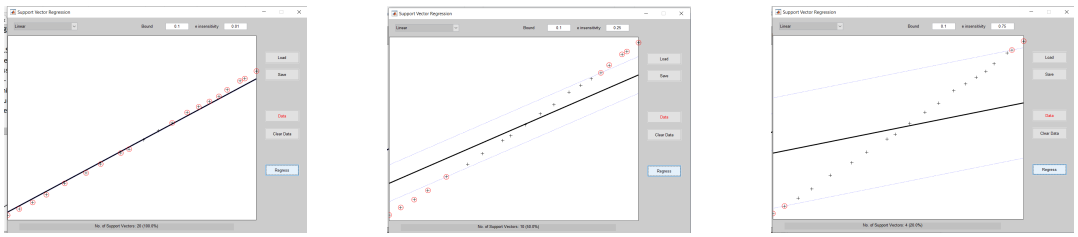


Figure 20: UI regress curve with constant bound and incresing e-sensitivity

Now we need to examine the influence of bounds on the model. In order to do that we fix the epsilon to 0.1 and we test with different values of Bound. This bound is associated with the VC dimension of SVMs . We select the an upper bound based on generalization error and use it for the selection of the hyperparameters. As we notice in the graph below when the bound is extremely small (0.01) we have a poor function estimation. The model hardly fits the data. Performance improves when we set the bound a little higher to 0.1 and we see evn better performance with bound . We obseve that in default setting with bound infinity the model fits the data perfectly. We get good results with larger bound (10 or 100) . But with higher bound there is a risk of overfitting. The optimal bound would be the lowest bound that just fit the data. Also when the bound is less , there are more number of support vectors and computation is slower whereas at higher bound the computation speed improves because of reduce in number of support vectors. This is associated with the sparcity of the model which refers to how well the model can perform with minimum number of the data points. sparcity of the model is used to arrive at optimal value of bound.

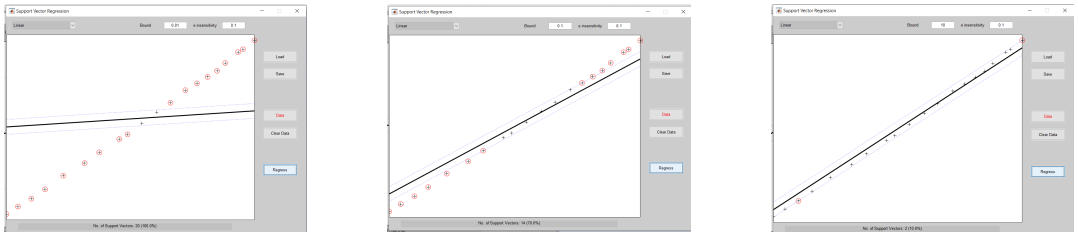
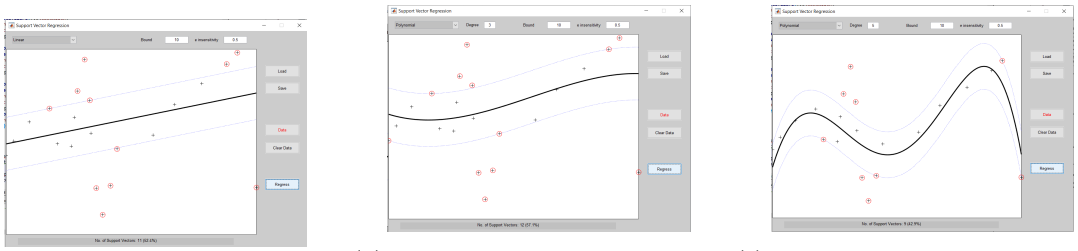


Figure 21: UI regress curve with constant e-sensitivity and incresing bound

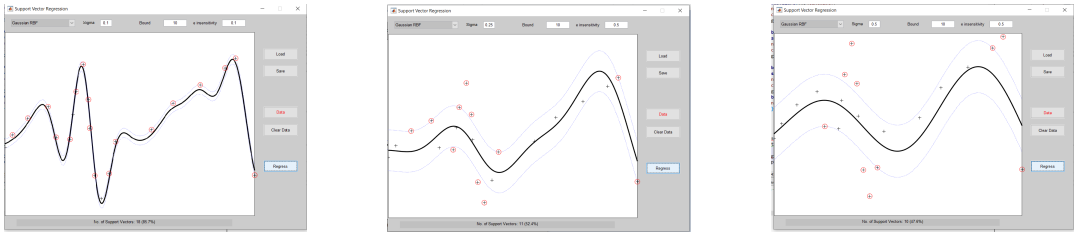
Question: Construct a more challenging dataset. Which kernel is best suited for your dataset . Motivate why.
To address this problem we create a more challenging dataset , which are not linearly regressible and has some outliers. We try to regress these points using RBF kernel and polynomial kernel and observe how well the model fits the data. We first try linear kernel(although we are pretty sure that linear kernel is not sutaible for such data) with e sensitivity 0.5 and bound 10. as per our expectation model doesnot fit the data well. Now we try polynomial kernel with same bound and e-sensitivity. We observe that for lower degree(3) the performance is poor whereas when we increse the degree to 5 performance improves and model regresses better. This is because our data is highly non linear and incresing that degree would mean introducing more non linearity to model , hence fits the data better.



(a) linear kernel (b) Polynomial kernel with degree 3 (c) Polynomial kernel with degree 5

Figure 22: Regression on complex dataset using linear and polynomial kernel

Now we try RBF kernel. Initially we keep the e-sensitivity low(0.1) and sigma as 0.1. We set the bound as 10 .We regress the complex data with these values and observe that model perfectly fits the data. It performs very well and probably overfits the data. 18 out of 20 i.e 85 percent datapoints are support vectors in this case so it exhibits minimal sparsity.So we increse the misclassification tolerance or e-sensitivity to 0.5 and sigma to 0.25.Now we obseve that model generalises better although it does not fits the data as good as previous setting. But this setting exhibits a better sparsity as 52 percent of datapoints are support vectors. Now we increse the sigma to 0.5 and notice that sparcity improve, 47 percent of data points are now support vectors but performance reduces.

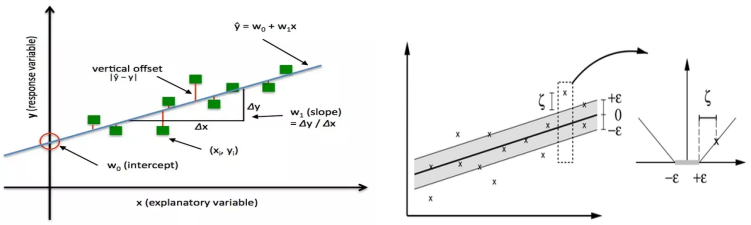


(a) RBF kernel (b) RBF kernel (c) RBF kernel

Figure 23: Regression on complex dataset using Gaussian RBF kernel

For this kind of complex dataset RBF Kernel is the most suited. Aprt from being flexible and it achieves good function estimation. RBF kernel gives good performance with non linear data We observe that for small sigma the curve is more flexible. If e-sensitivity is small we achive a good function approximation.Bound = 1 is a good compromise as it sellets the balance between generalization and performance.

Question :In what respect is SVM regression different from a classical least squares fit?
In classical least square regression, we use squared loss function in order to find the line that fits the best. Squared loss function finds a line with minimum distance from the observations. Where as Linear-Support vector regression, we use epsilon insensitive L1 loss function, which means if data has error are within the threshold of epsilon then the error is tolerated i.e. error is considered as zeror. else considered error e1 = actual error - epislon. Hence observation that are outside the epsilon range produces error.



(a) least square fit (b) SVM regression

Figure 24: Image source : INTERNET

As illustrated in figure 24 b with the visualization of the episolon tube of the function estimation , All observation that are support vectors are included in the shaded area and the error of all those points is equal or smaller to episolon hence the error is considered as zero.

2.2 A simple example: the sinc function

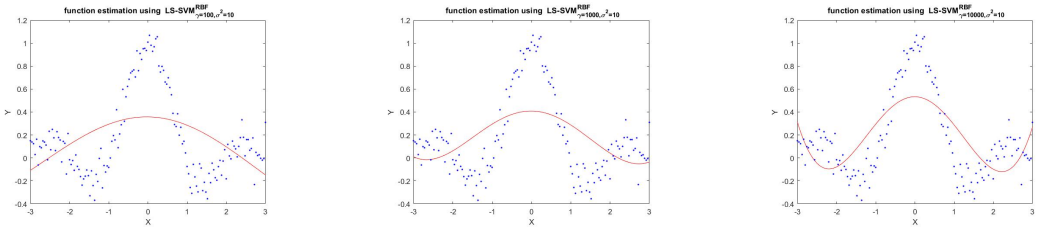
2.2.1 Regression of the sinc function

Question : Try out a range of different gam and sig2 and visualize the resulting function estimation on the test set data points. Discuss the resulting function estimation. Report the mean squared error for every combination (gam, sig2).Do you think there is one optimal pair of hyperparameters?
Here we try to gain insight about function estimation .We construct artificial dataset from the sinc function with white noise and apply the LS-SVM model for Function Estimation.We calculate Mean square error for each combination and plot Ytest against estimated Ytest.
MSE = $(ytest - ytest\_estimated)^2$/number of ytest We experiment with 3 different gamma and different sig2 values and tabulate the MSE values for the resulting 16 combinatons of gamma and sig2 We use following values og gamma and sig2:
gamma gam = [10 100 1000 10000];
sig2 = [0.01 0.1 1 10]

gam/sig2	**gam=10**	**gam=100**	**gam=1000**	**gam=10^4**
0.01	1.144	1.14	1.15	1.17
0.1	1.07	1.07	1.09	1.11
1	3.808	1.88	1.27	1.07
10	11.68	11.32	9.6	6.89

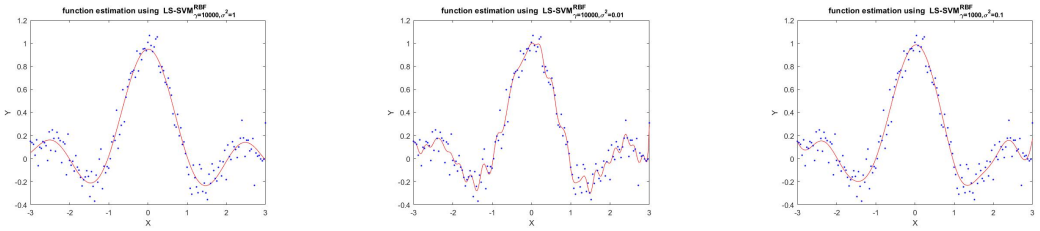
Table 5: MSE for different combinations of gamma and sig2

From the table we learn that low value of sigma and gamma results in good fit.As sigma/gamma both increses the mean squared error increses. we get highest mean squared error for sig2=10. The optimal combination is gamma=10 and sig2=0.1 as it gives the least mean squared error(1.07).Also gam =100 and sig2=0.1 gives the best fit of all.In figure 25 we see the plot between ytest and y estimated for non optimal combinations of gamma and sig2. We observe that there is huge variations between the estimated output and actual output.These combinations give high mean squared error



(a) gam=100,sig=10 (b) gam=10000,sig=10 (c) gamma=10000, sigma = 10

Figure 25: plot between ytest and y estimated for non optimal combinations of gamma and sig2



(a) gam=10,sig=.1 (b) gam=10000,sig=1 (c) gamma=1000, sigma =.10

Figure 26: plot between ytest and y estimated for optimal combinations of gamma and sig2

While for optimal combination of the gamma and sig2 we observe that estimated and actual output almost overlaps. Key observation here is MSE is low for lower values of sig2. Optimal value of sig2 is 0.01 to 1.

Question : is there a optimal pair of hyper params?

From the above table we get a combination that appears to be optimal since it gives the least MSE. but that happens to be a local optimal point. There is no single combination that can be declared as optimal combination ,rather a range of combination. Moreover on multiple runs the most optimal combination shifts within this range. in our case optimal range of gamma is 10 to 10^4 and for sig2 0.01 and 10.

Question : 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 We tune the gamma and sig2 using tunlsvm and run for 5 iterations and get the following values. We use both simplex and gridsearch. We calculate the time taken using tic-toc and as expected grid search taked almost twice the time

Simplex			Grid search		
gamma	**sig2**	**error**	**gamma**	**sig2**	**error**
14299	2.3549	1.06	25.076	1.0537	1.110703
20.156	0.9105	1.04	57.19	1.4352	1.089782
25.656	1.1621	1.07	20.385	0.77856	1.052441
77.633	1.30278	1.05	64.897	1.2205	1.080197
2.0914	0.31947	1.06	25517	2.7546	1.095616

Table 6: tuned pair of parameter

On the genrated tuned pairs of si2 and gamma we find the error using the previous procedure and observe that we do get least error with the generated pairs . We get low errors on genrated pairs by both algorithms however average error of simplex pair is little less than grid search . Moreover simplex is almost twice as faster as gridsearch

2.2.2 Application of the Bayesian framework

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

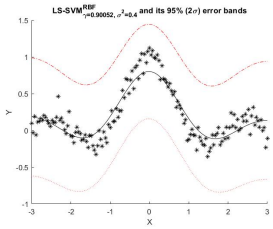


Figure 27: gamma=0.4, sigma =10

Bayesian framework works in three levels:

- *Level 1* [inference of parameters w, b]

$$p(w, b|\mathcal{D}, \mu, \zeta, \mathcal{H}_\sigma) = \frac{p(\mathcal{D}|w, b, \mu, \zeta, \mathcal{H}_\sigma)}{p(\mathcal{D}|\mu, \zeta, \mathcal{H}_\sigma)} p(w, b|\mu, \zeta, \mathcal{H}_\sigma)$$

- *Level 2* [inference of hyperparameters μ, ζ]

$$p(\mu, \zeta|\mathcal{D}, \mathcal{H}_\sigma) = \frac{p(\mathcal{D}|\mu, \zeta, \mathcal{H}_\sigma)}{p(\mathcal{D}|\mathcal{H}_\sigma)} p(\mu, \zeta|\mathcal{H}_\sigma)$$

- *Level 3* [inference of kernel parameter σ and model comparison]

$$p(\mathcal{H}_\sigma|\mathcal{D}) = \frac{p(\mathcal{D}|\mathcal{H}_\sigma)}{p(\mathcal{D})} p(\mathcal{H}_\sigma)$$

Source: course document

The objective of Bayesian framework is to maximize of the posterior distribution, given the prior distribution and data. We also need is to minimize the w vector which is related to the data and the sum of squared error evaluated on the training data. Both parts are multiplied with the hyperparameters which consist of regularization constant. The Bayesian framework in LS-SVM works has 3 levels. In the first level we perform inference of parameters w and b to approximate the function. In the second level, we maximize the log of posterior to get optimal pair of hyperparameters μ and ζ, where $\gamma = \zeta/mu$ is the regularization term. In the third level we do inference for kernel.We perform model comparison based on the evidence. Here are the steps for parameter tuning in Bayesian inference:
1) Inputs should be normalize to standard value i.e. zero mean and 1 variance.
2) We choose a kernel(RBF , Linear , Polynomial) in order to select a model
3) Compute the effective number of parameters from the eigenvalued decomposition of the centered Gram matrix.
4. Find the optimal hyperparameters μ and ζ to solve the optimization problem $\gamma = \zeta/mu$ to to maximize the Level 2 posterior.
5.Compare the model based on evidence
6. Refine the tuning params of the kernel and jump to step 2 till model improvises.

In the case of RBF kernel input selection along with inference of kernel parameter can be done at third level. with Automatic Relevance determination. For linear or polynomial kernels, we dont need third level. In the code, where the parameters [alpha, b], gamma and selection of sig2 are done at level 1,2 and 3 respectively.If we take the value of gam and sig2 generated by tunelsvm(gamma = 2.0914 and sig2 = 0.31947) and calculate the MSE(MSE=0.0467) and compare it with mse of default setting(gamma=0.4 and sig2 = 10 MSE=0.198) we infer that tuned pair results in less error.

2.3 Automatic Relevance Determination

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

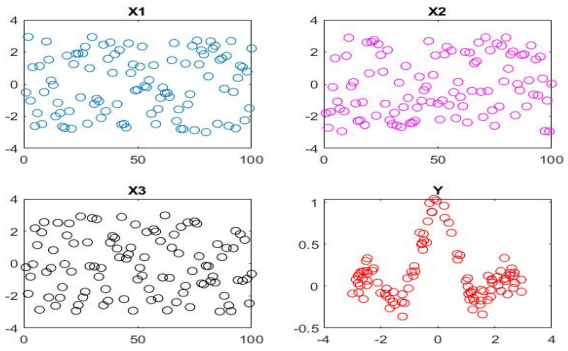
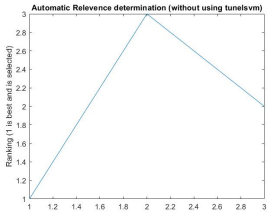
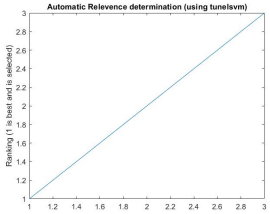


Figure 28: Data distribution X1,X2andX3 and Y

Given above is the data distribution of 3 input variable X1 ,X2, X3 and Y



(a) Rank plot Without using tunelsvm(gamma=4 and sig2=10)

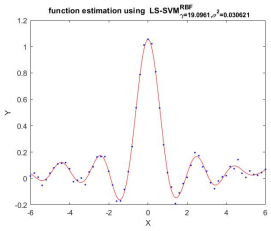


(b) Rank plot using tunelsvm

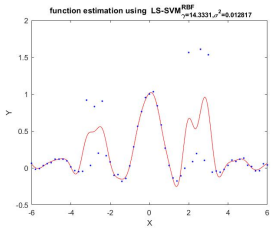
For random pair of gamma and sigma[gamma=4 and sig2=10] the rank sequence generated is [1,3,2].This means that first term is most relevant followed by third.This is relative relevance not absolute.We select input using the crossvalidation under tunelsvm. In this way the tuned gamma and sig2 will be selected(gam= 76.8 , sig2 =0.58) When we generate gamma and sig2 using tunelsvm the rank sequence is [1,2,3] i.e first term is most relevant followed by second then third.

2.4 Robust Regression

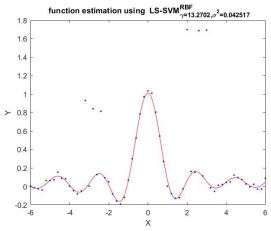
Question : Visualize and discuss the results. Compare the non-robust version with the robust version. Do you spot any differences?



(a) Data without outlier(so no robust regression required)



(b) data with outlier without robust regression



(c) data with outlier Using Robust regression

When the data is corrupted with non-Gaussian noise or outliers, its important important to incorporate robustness to the function estimation.Inorder to infer the same we create a data and add outliers to it.We first train a LS-SVM regressor model, without giving special attention to the outliers.Then train a robust LS-SVM model, using robust crossvalidation.First figure is generated without using the outliers so even though the robust regression is not used the function estimation is good.In second figure robustness is not applied because of which outliers in the dataset have a big impact on the model resulting into a worst function estimation. In seconcd figure we apply robust regression,the noise is ignored by the model and a better function a=estimation is achieved. Model clearly fits the function sinc (X) + 0.1.* rand (size (X)) and effect of outliers is nullified. The above illustration verifies the role of Robust regression in presence of outlier.

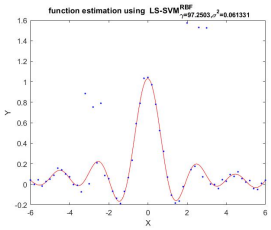
Question : Why in this case is the mean absolute error ('mae') preferred over the classical mean squared error ('mse')?

When the MSE is used the effect of outliers is magnified since we take squared distance here. If the outliers is far from the function the square is a large number and hence the error. If we take absolute values of Error so the outlier have less influence on model.

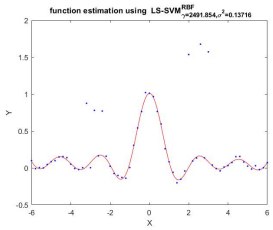
Try alternatives to the weighting function wFun (e.g., 'whampel', 'wlogistic' and 'wmyriad'. Report on differences.

	Huber	**Myriad**	**Logistic**	**Hampel**
No of iterations to converge	20	26	39	5
Gamma	97.2503	2491	23.9	114
sig2	0.0613308	0.137	0.049	0.06

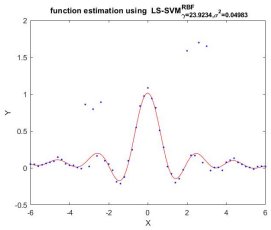
Table 7: tuned pair of parameter



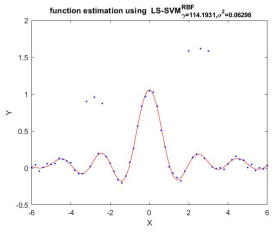
(a) Weighting function:Huber



(b) Weighting function:Myriad



(c) Weighting function:Logistic



(d) weighting function: Hamphel

We apply 4 weighting functions:Huber,Myriad,Logistic and Hampel. We achieve robust performance and tuned pair of hyperparameters from all function.However we observe a slight influence of otliers on myriad,logistic and liitle bit evn in hamphel. Huber seems to give nearly perfect function estimation. Huber is less sensitive to outliars . Both huber and hamphel are sparse in nature. We also observe that hamphel takes shortest to converge, only 5 iterations followed by huber (20 iterations). We also observe that MAE obtained for all the weighting function is almost same i.e 0.134 approximately.However Myriad huber has least MAE.

2.5 Homework

2.5.1 Logmap Dataset

Question: As indicated numerous times before, the parameters gam and sig2 can be optimized using crossvalidation. In the same way, one can optimize order as a parameter. Define a strategy to tune these 3 parameters. Do time series prediction using the optimized parameter settings. Visualize your results. Discuss.
We implement the code with default values of gamma= 10, sigma=10 and order = 10, we get the following prediction. It performance doesnot seem great. while few spikes matches others show ofset, we can see huge ofset near deep peaks. Few of the spikes are accurate

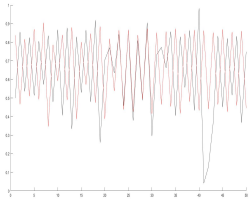
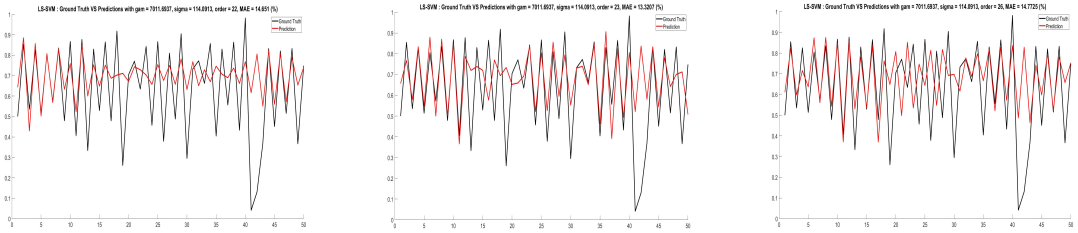


Figure 32: Graph depicting actual and predicted data with gamma=10 and sig2=10, order=10

Now we tune the values of gamma , sigma and order to obtain minimal error. we try to select best order from range 5 to 50 .Here are some best prediction graph with least error.



(a) gamma=7011 ,sig2=114 ,order = 23

(b) gamma=7011, sig2= 114,order=22

(c) gamma=7011 ,sig2=114.01, order=26

Figure 33: Graph depicting actual and predicted data with tuned values of gamma and sigma

We can crossvalidation to tune the parameters gamma, sig2 and order.
We tune the value of sig2 and gamma using tunelsvm and to get the best value of order we select few order and run the algorithm in a loop and check the error for each combination and select the order with least error. In our case order = 23 gives least error of around 13 percent. We observe that we obtain great accuracy at the beginning, however this accuracy worsens in the middle and deep spikes but picks up later. Also the best pair of gamma and sigma selected by tunelsvm (gamma=7011 and sig2=114) Initially the actual and precicted peaks seem to coincide , however we observe ofset in the middle which increses further. we could not achive a great accuracy even with tuned values and error doesnot drop below 13 percent.

2.5.2 Santa-Fe data

Question: Does order = 50 for the utilized auto-reressive model sounds like a good choice?
Order in auto-regressive modeling is number of periods apart or the lag the prediction and actual groundtruth data .If order is 50, the model will make a prediction specific point which lags the actual groundtruth by 50 points.This difference is huge. It is better to choose a lower value or order. .

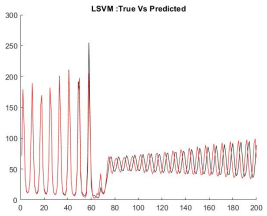
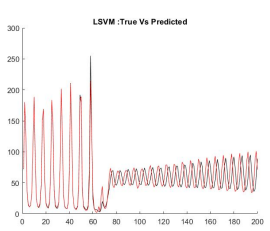
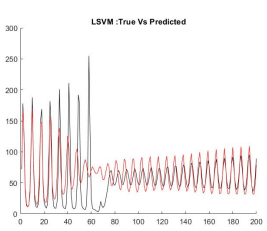


Figure 34: Graph depicting actual and predicted data when order is 50, gamma= 50, sig2=10

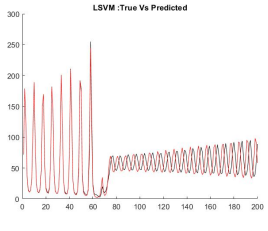
Question :Would it be sensible to use the performance of this recurrent prediction on the validation set to optimize hyperparameters and the model order? Tune the parameters (order, gam and sig2) and do time series prediction. Visualize your results. Discuss.
Yes it would make sense to use performance of recurrent prediction pn validation set to optimize hyperparameters and order since in this case we will be getting autocorrection of values. Use of validation set helps in minimization of error in each iteration resulting in optimization of hyperparameters.



(a) gamma=1.3111e+03 ,sig2= 4.0920 MAE=0.00790 ,order = 14



(b) gamma=24.2410, sig2= 54.9,MAE=0.1934,order=28



(c) gamma= 3.6932e+03,sig2=7.21,MAE=0.073 order=35

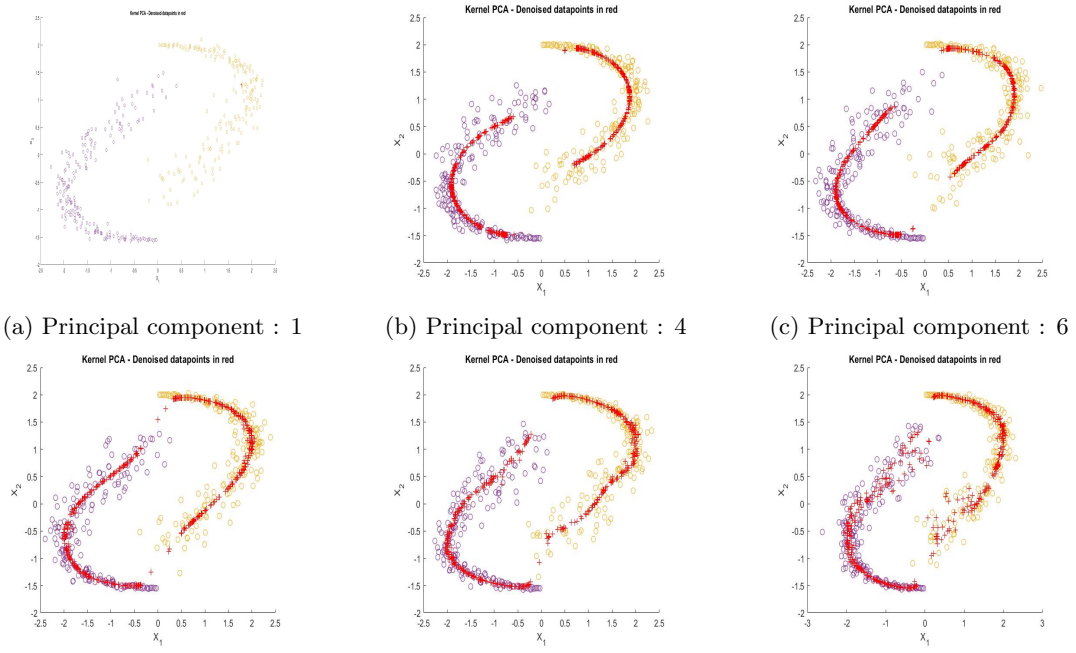
Figure 35: Graph depicting actual and predicted data with tuned values of gamma and sigma

We try various combination of gamma , sig2 and order. We choose gamma andsig2 using tunelsvm and pick order from range 5 to 40. we calculate MAE on each combination and pick the ones with least MAE and best performance. We get the above visualization for best picked combination of gamma , sig2 and order. We observe that we get great predictions for the first part of the time series as true and predicted values coincide and the performance decreses as ofset beginsto decline for the second part of the data.

3 Exercise session 3 : Unsupervised Learning and Large Scale Problems

3.1 Kernel principal component analysis

Question: Describe how you can do denoising using PCA. Describe what happens with the denoising if you increase the number of principal components.
Principal Component Analysis is a data transformation technique where principal component are the linear combination of predictors which could be ordered by their Eigenvalue. If the Eigenvalue is big it means more variance is covered. PCA transforms the data by laying the features space to the lower dimention so that greatest varince (first principal component) lies in first coordinate and second greatest variance lies in second coordinate .It performs an orthogonal linear transformation in order find a projection of all data into k dimensions. Denoising of data carried out in the regions where the dimentions of noise and features are different. We run kpca script with 400 data points with added noise. To acheive the reconstruction we use 1,4,6 , 8, 10 ,12 principal component[RBF kernel (sig2 =.4), Approximation technique : Lanczos] as shown below.



(a) Principal component : 1 (b) Principal component : 4 (c) Principal component : 6

(d) Principal component : 8 (e) Principal component : 10 (f) Principal component : 12

Figure 36: denoising effect when we increse no of principal component

When we increse the number of principal components we preserve more information from the input space, therefore, while denoising of images, the reconstructed images are less blured and closer to the original ones,with the noise being removed. However after incresing till certain amount it begins to overfit.

Compare linear PCA with kernel PCA. What are the main differences? How many principal components can you obtain?
The linear PCA finds linear principal components to represent the data in lower dimension.If the standard PCA is applied for this data, we wont be able to find good representative direction. Kernel PCA rectifies this limitation. Kernel PCA uses Kernel trick in order to find principal components in a higher dimentional space. Moreover computational time for KPCA to extract principal components more compared to Standard PCA. We notice That we need a minimum of 3 principal component for minimum performance.For linear PCA, the number of principal components is equal to the number of dimensions of the input space. For Kernel

PCA, it is possible map the input space to a higher-dimension, therefore the no of principal components can be larger than the dimension of the input space , it can be as high as number of data points.

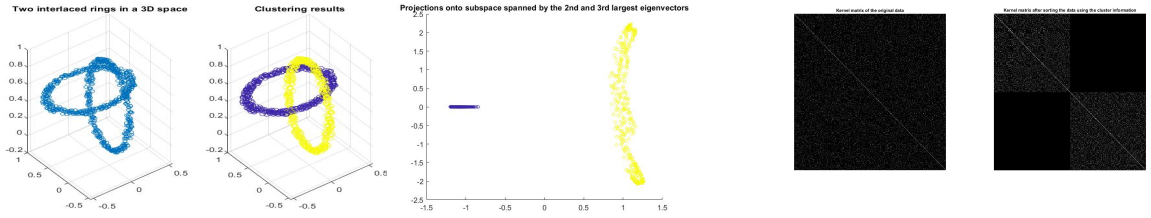
Question :For the dataset at hand, propose a technique to tune the number of components, the hyperparameter and the kernel parameters.

In order to determine the best selection of no of principal components, we map the eigenvalues in to evaluate which are the largest ones and in where they gradually decline. Sometimes, optimal number of principal components that we need to choose can be simply inferred from the graph. In cases where it is not obvious, we try different numbers of principal components and evaluate the reconstruction error. Moreover we can also evaluate the performance on unseen data points. We can finetune the Kernel hyperparameters gamma and sigma by Cross-validation.

3.2 Spectral clustering

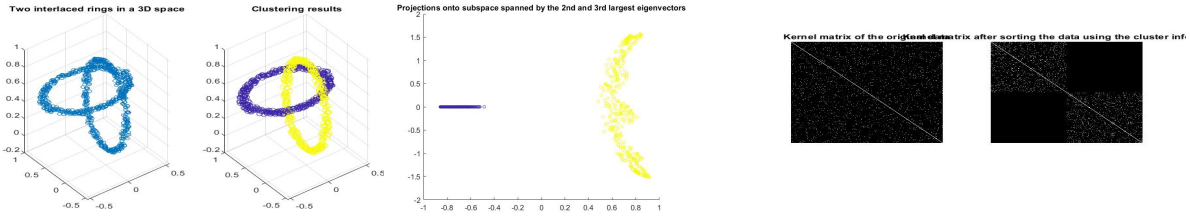
Question: Explain briefly how spectral clustering works .What are the differences between spectral clustering and classification?
In spectral clustering ,data points acts as nodes of a connected graph and clusters are found by partitioning this graph, based on its spectral decomposition, into subgraphs.
Process of spectral clustering can be summarised into 3 steps :
1)First a similarity graph for all data points is constructed(eg KNN graph)
2) Embed data points to a low dimentional space where clusters are more obvious with the use of eigen vectors (Known as spectral embedding).we need to compute the first k eigenvectors of its Laplacian matrix to define a feature vector for each object.
3) Classical clustering algorithm is applied to partion the embedding The basic difference between classification and spectral clustering is , classification is a supervised learning where learning happens on labelled training data and tested on unseen data ,whereas Spectral clustering is an unsupervised learning method, where the model is fed with unlabeled data and tries to recognize patterns of the data itself. The second difference is the In clustering we have additional weighting term v for each point's error we introduce to the model, which contain information about the closeness between the data points.

Question : Edit the script and try different values of sig2 (e.g., 0.001, 0.005, 0.01, 0.02). What is the influence of the sig2 parameter on the clustering results?
For the lower values of sigma (0.001, 0.005 and 0.01) we get good results as the two patterns get assigned correctly to their own cluster. In eigenvector space the 2 clusters are linearly separated.For higher values of sigma (eg. 0.02 or 0.5) variance of the eigenvectors increases, thus they are not linearly separable and rings dont get clustered properly.



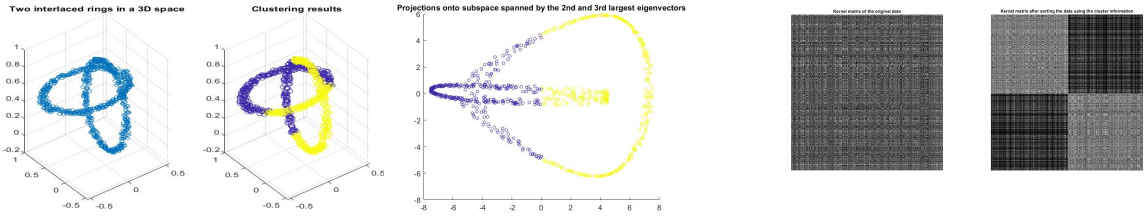
(a) Clustering result (b) eigen vector space (c) Kernel Matrix of original and clustered Data

Figure 37: Clustering with sigma = 0.01



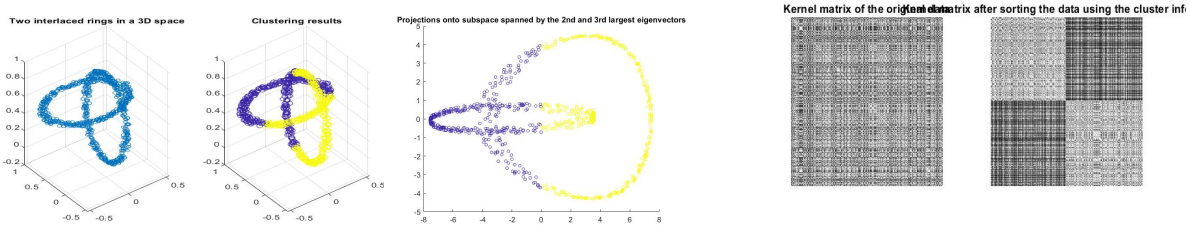
(a) Clustering result (b) eigen vector space (c) Kernel Matrix of original and clustered Data

Figure 38: Clustering with sigma = 0.005



(a) Clustering result (b) eigen vector space (c) Kernel Matrix of original and clustered Data

Figure 39: Clustering with sigma = 0.2



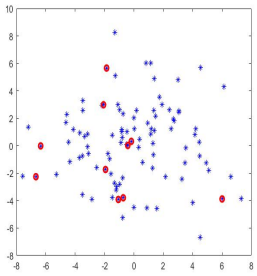
(a) Clustering result (b) eigen vector space (c) Kernel Matrix of original and clustered Data

Figure 40: Clustering with sigma = 0.5

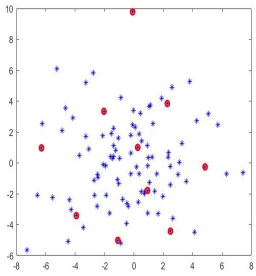
3.3 Fixed size LS-SVM

Question : In which setting would one be interested in solving a model in the primal? In which cases is a solution in the dual more advantageous?
Primal would be interesting for large size data problems because in the primal, the unknowns in the objective function are the w vector, which depends on the dimensionality of the input space, and the bias term. whereas For large dimentional input its advantageous to use dual representation.

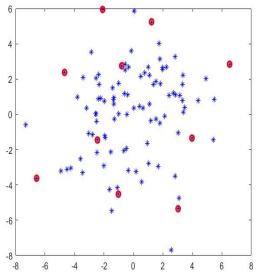
Question : What is the effect of the chosen kernel parameter sig2 on the resulting fixed-size subset of data points (see fixedsize script1.m)? Can you intuitively describe to what subset the algorithm converges?



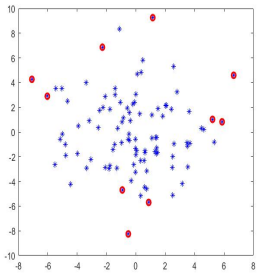
(a) sig2=0.001



(b) sig2=0.01



(c) sig2 = 1

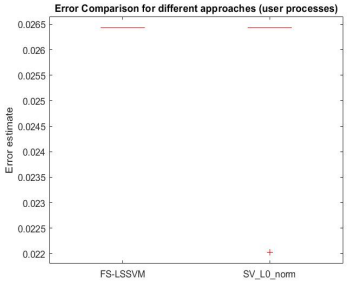


(d) sig2 = 100

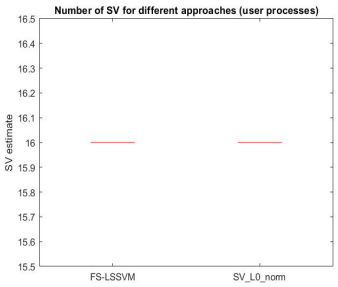
Figure 41: Effect of sig2

There are 2 ways we can select subset : Random and quadratic Renyi entropy. We first initiate the selection of subset at random by picking one point of training data. That point is considered as a candidate to become a support vector in the working set. It will be chosen if it is proven that it improves the entropy criteria, else gets rejected. We repeat the process with the next data point till convergence. The Renyi entropy quantifies the diversity of the subset in to ensure that the subset has a good representation of the entire dataset.which is acquired when sigma is equal to 1. When sigma value is reduced (0.01,0.001) Subset is chosen at random. When sigma equal to 100, the points of the subset is at the edges of the dataset,includes outlier points as well sometimes. Basically when sigma increses points tens to spread out.

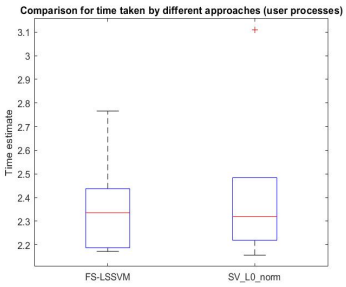
Question : Run fslssvm script.m. Use the Wisconsin Breast Cancer dataset for this exercise. Compare the results of fixed-size LS-SVM to 'L0-approximation in terms of test errors, number of support vectors and computational time.



(a) Error comparison of different approches



(b) number of support vectors



(c) Comparison of time taken

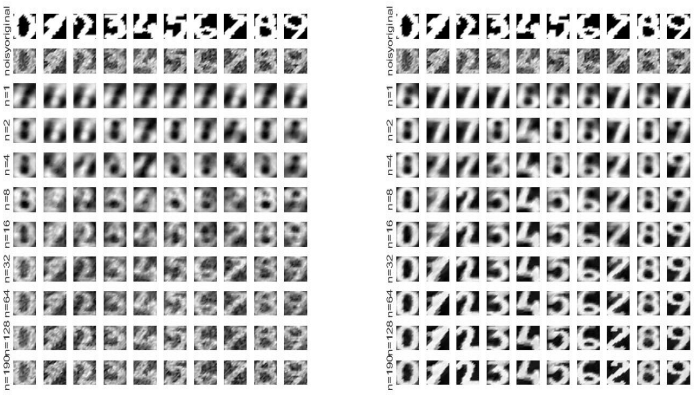
Figure 42: Fslsvm_script on Wisconsin Breast cancer dataset

Here we compare FS LS-SVM and lo approximator. In this case in terms of error level the two models do not show any difference and it comes around 0.0265.Also both LSSvm and Lo approximator takes same number of support vectors as Lo approximator i.e 16. But when it comes to computational FS-Lsvm takes more computational time than Lo approximator. At the same time Fs-lsvm show higher variablity too.

3.4 Homework

3.4.1 Kernel Principal Component Analysis

Question :Illustrate the difference between linear and kernel PCA by giving an example of digit denoising for noisefactor = 1.0. Give your comments on the results (based on visual inspection).



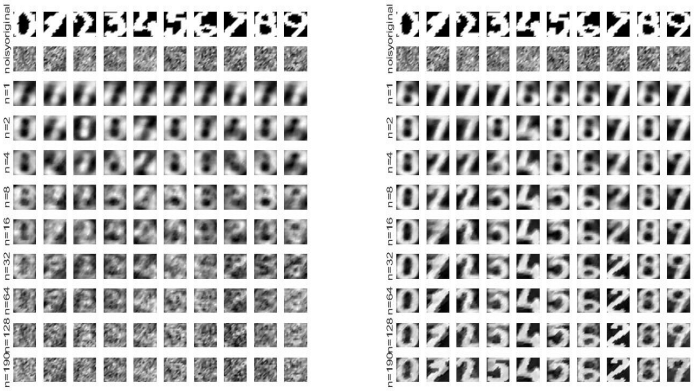
(a) Linear PCA, noisefactor = 0.3, sig2 = 0.7

(b) RBF kernel PCA, noisefactor = 0.3, sig2 = 0.7

Figure 43: Comparison between Linear and kernel PCA with noise factor =0.3

We first run the script digitsdn with the default parameter noise factor 0.3 and sig2 = 0.7 . We observe that kernel PCA performs better than linear since it can remove the noise better and original inputs is reconstructed in a good level except 7 (which some how resembles 2 and reconstructed as 2). Linear PCA reconstructruct the original input digits but the reconstruction is still noisy. Now we increse the noise to 1 keeping sig2 same. As expected noise levels is high and digits are unreadable. Linear PCA performs poorly, it fails to remove noise and reconstruction achived after 190 principal component are also not properly readable. The Kernel PCA removes the noise to satisfactory level after using 64 Principal components. In this case too digit 7 cannot be reconstructed properly. After using 190 principal component the reconstruction has distortions.The reconstructed digits one, three, five, seven and nine is completely distorted as we can see in figure below. Results can be improved by tuning sig2

.



(a) Linear PCA, noisefactor = 1, sig2 = 0.7

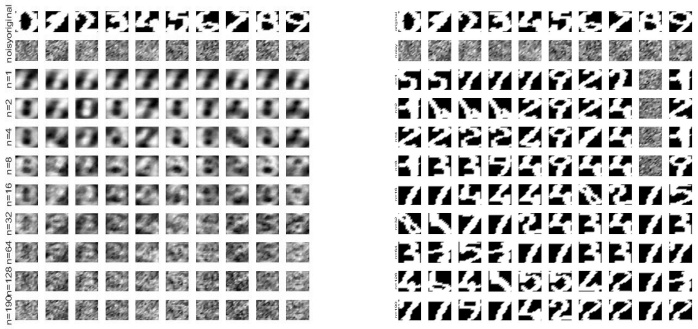
(b) RBF kernel PCA, noisefactor = 1, sig2 = 0.7

Figure 44: Comparison between Linear and kernel PCA noise factor =1

Question : What happens when the sig2 parameter is much bigger than the suggested estimate?What if the parameter value is much smaller? In order to investigate this, change the sigmafactor parameter for equispaced values in logarithmic scale
For smaller the sig2 value, the priority is denoising of the image, and less focus is given on the minimization

of the reconstruction error.So when we see sig2 = 0.01 we get all the reconstruction digits wrong for both linear and Kernel PCA. For linear PCA reconstruction is wrong and blurry but for RBF reconstruction is only wrong.

.



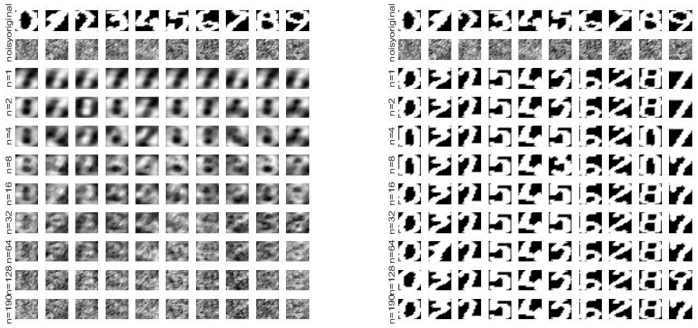
(a) Linear PCA, noisefactor = 1, sig2 = 0.01

(b) RBF kernel PCA, noisefactor = 1, sig2 = 0.01

Figure 45: Comparison between Linear and kernel PCA noise factor =1 and sig2=0.01

When we increse the sig2 a bit to 0.1 we observe that linear PCA still gives poor and blurry results but results of RBF kernel begins to improve. We observe that reconstruction of digit 0,2,4,6,8 happens correctly after 1st principal componen. Digit9 reconstructs properly at 128th principal componet but distorts at 190 as we can see in figure below.

.



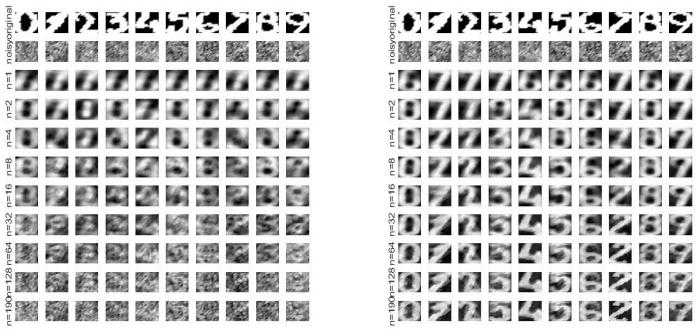
(a) Linear PCA, noisefactor = 1, sig2 = 0.1

(b) RBF kernel PCA, noisefactor = 1, sig2 = 0.1

Figure 46: Comparison between Linear and kernel PCA noise factor =1 and sig2=0.1

When we increse the sigma to 1 we obseve there is no significant improvement in performance of linear PCA but Reconstruction of RBF improve but results are blurry. so noise removal is decresed. We observe that except 1 and 7 all other digits are reconstructed properly.When we further increse the sig2 to 10 ,still no significant change in performance of Linear PCA but now RBF PCA behaves extremely poorly , and its performance is almost as same as linear PCA. Less sigma helps in noise removal. High sigma helps in digit reconstruction. so we need to find a optimal value of sigma where both the purpose is served equally.

.

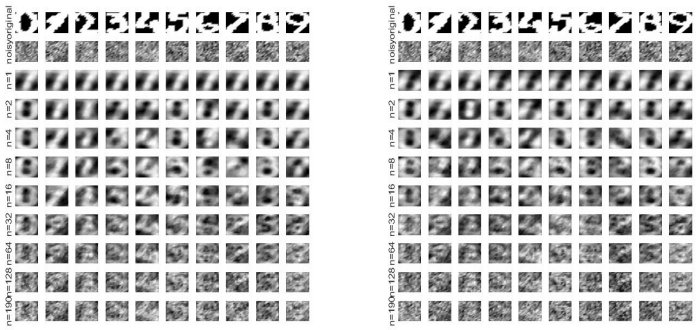


(a) Linear PCA, noisefactor = 1, sig2 = 1

(b) RBF kernel PCA, noisefactor = 1, sig2 = 1

Figure 47: Comparison between Linear and kernel PCA noise factor =1 and sig2=1

.



(a) Linear PCA, noisefactor = 1, sig2 = 10

(b) RBF kernel PCA, noisefactor = 1, sig2 = 10

Figure 48: Comparison between Linear and kernel PCA noise factor =1 and sig2=10

Question: 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?
We observe that the reconstruction error on all three data set (Xtest, Xtest1 and Xtest2) is quite high when we run Linear kernel. So we select RBF kernel for this experiment. In this case we observe that for less value of sigma(0.001) reconstruction is extremly poor and reconstruction error is high. For very high value of sigma(40) , although the reconstruction error is expected to be low but because of noise interference the reconstruction is poor. we observe considerable amount of reconstruction error with mid values of sigma, say 1 or 10.Less sigma helps in noise removal. High sigma helps in digit reconstruction

3.5 Fixed-size LS-SVM

3.5.1 Shuttle (statlog)

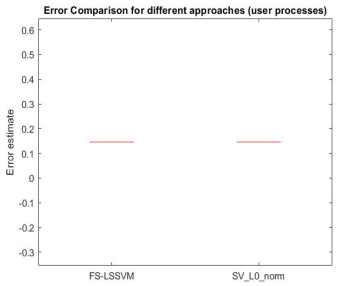
Question : Explore and visualize (part of) the dataset. How many datapoints? How many and meaning of attributes? How many classes? What is to be expected about classification performance
Shuttle(Statlog) dataset has 700 data points and 10 attributes(X is 7X10), where 9 are the input variables

and 1 is the class. When we visualize Y vector with the characterization of the classes, we know that this is a multi-class classification problem, 78 percent of data belongs to 1st class, Lesser amount of data comprise of 4th and 5th class while the 2nd and 3rd class are characterized by just sufficiant amount of data points. Class content is tabulated below.

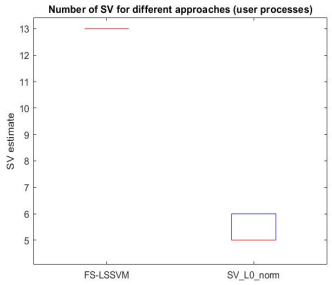
Class	**Num Data points**	**percentage of total data points**
1	45586	78.60
2	50	0.09
3	171	0.29
4	8903	15.35
5	3267	5.63
6	10	0.02
7	13	0.02

Table 8: Data discription of Shuttle

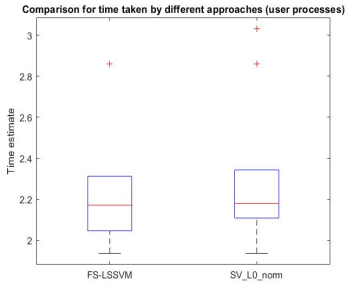
Since the data points is extremely unbalanced Classification results might be over fitted.
Question : Visualize and explain the obtained results.



(a) Error comparison



(b) number of support vectors



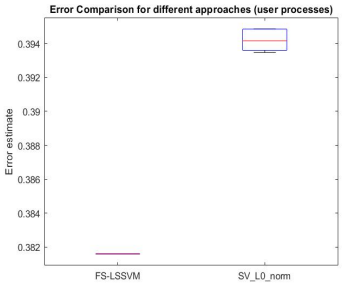
(c) Comparison of time taken

Figure 49: Fslsvm_script on Shuttle dataset

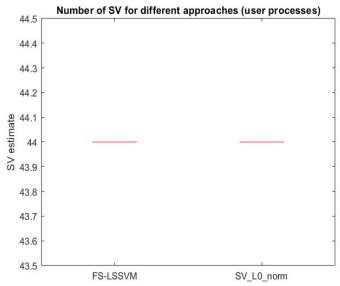
To visualize the results we run the Fixed size LSSVM on shuttle . we observe that error level is on the same level i.e 0.15 for both FSLSVS and lo method, FSLS-SVM uses more Support Vectors (13) than Lo estimator. The time needed for training the LS-SVM model is longer compared to the Lo estimator also it has higher variablity and this can be explained due to the fact that the algorithm goes through an iteration process evaluating all the data points as potential Support Vectors and finally accepts or rejects them according the Renyi entropy criteria

3.5.2 California

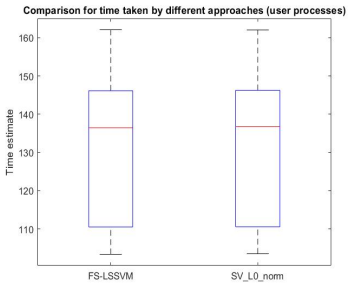
Question : Explore and visualize (part of) the dataset. How many datapoints? How many and meaning of attributes?
The California dataset is a huge dataset, as it consist of information about housing in California state since 1990 and contains 20640 data points and 9 different variables like median income, house age, no of rooms, no of bedrooms, population, households etc.



(a) Error comparison



(b) number of support vectors



(c) Comparison of time taken

Figure 50: Fslsvm_script on California dataset

We run the script to visualize the results. And in this case unlike in previous datasets FS LSSVM performs much better than Lo estimation model because the error is very less(less than 0.382). Here we can see the advantage of FSLSSVM for large scale data because previous data set size was much much smaller. Number of support vectors required for both the methods is 44 and computational time tis also similar. Also for both case computational time shows variablity.