Kernel Method for Pattern Analysis

CS6011

IIT Madras

Assignment 3

By:
Group 6:
Arun Baby (CS15S016)
Vishal Subbiah (MM12B035)
Nikhil Thomas Stephen (CS15M032)

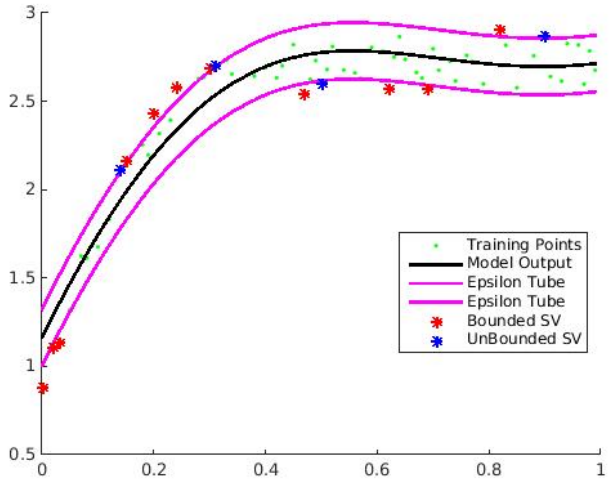
Contents

1 Regression

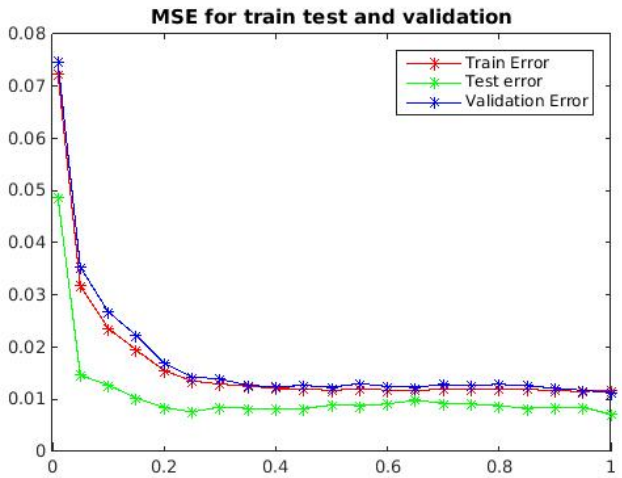
1.1 Dataset 1: 1-D input data

1.1.1 Observation and Inference

- From fig 1b we can see, as ν value increases the error on the training, validation and test data decreases.
- The settings used are ν =0.2, γ=1 and C=25.
- Among the different methods, polynomial curve fitting gave the best approximation for the undelying function.

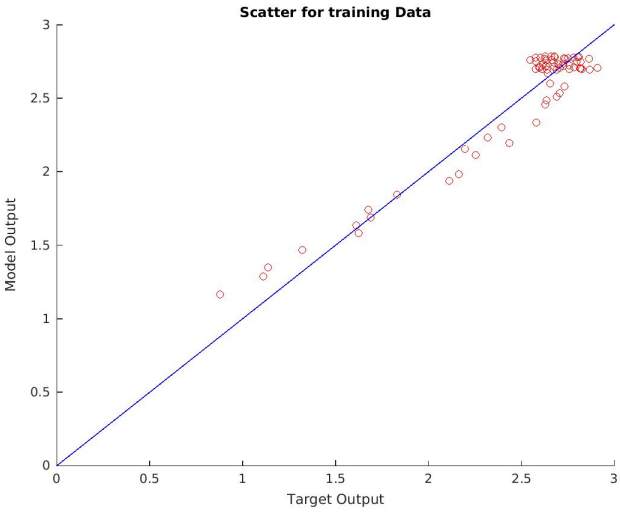


(a) Approximated Function with ϵ tube

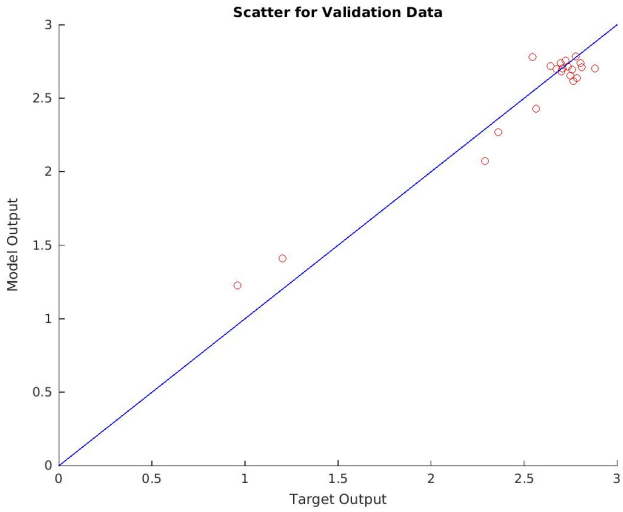


(b) Variation of Error with respect to ν values

Figure 1: Using 66 training data points



(a) Scatter plot for Training Data



(b) Scatter plot for Validation Data

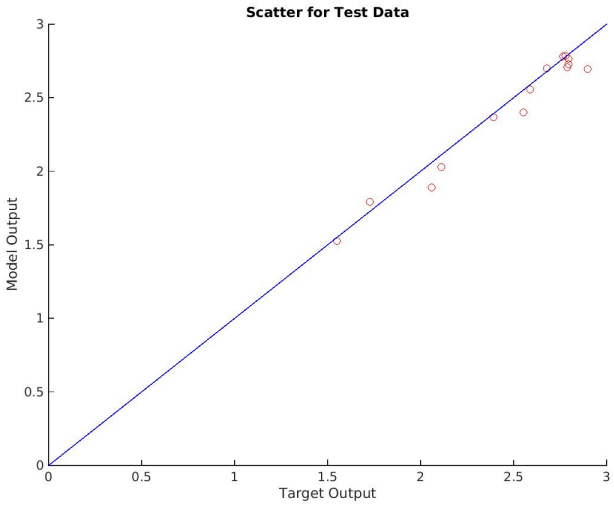
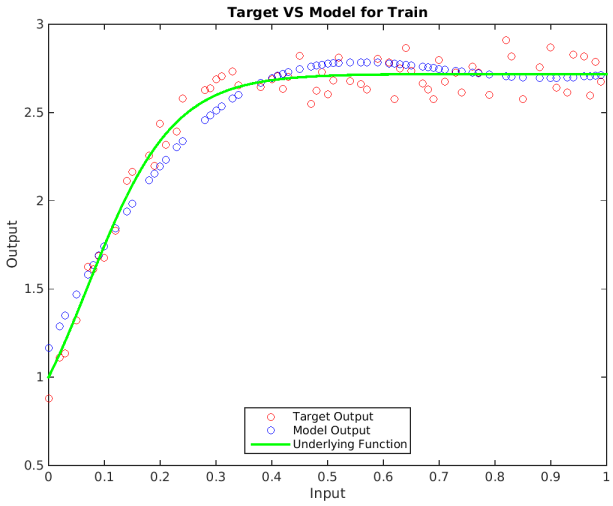
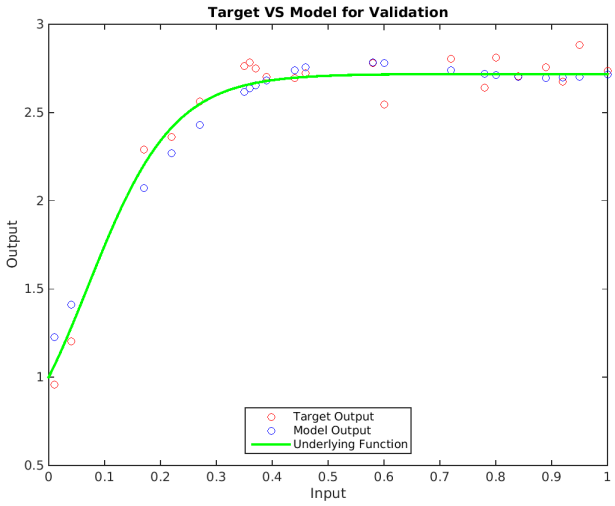


Figure 2: Scatter plot for test data



(a) Target vs Model for Train



(b) Target vs Model for Validation

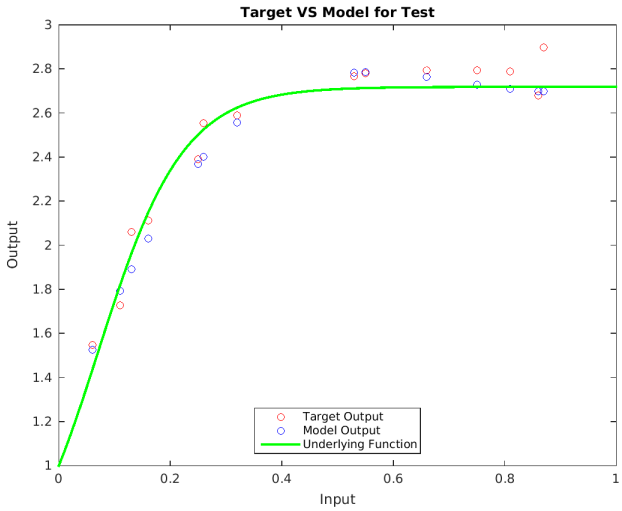
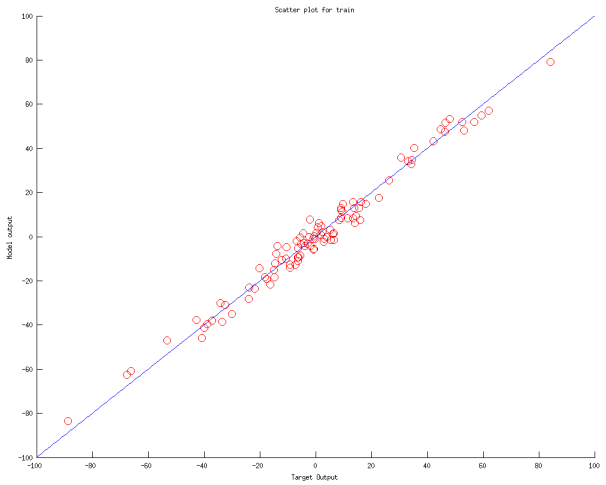


Figure 3: Target vs Model for Test

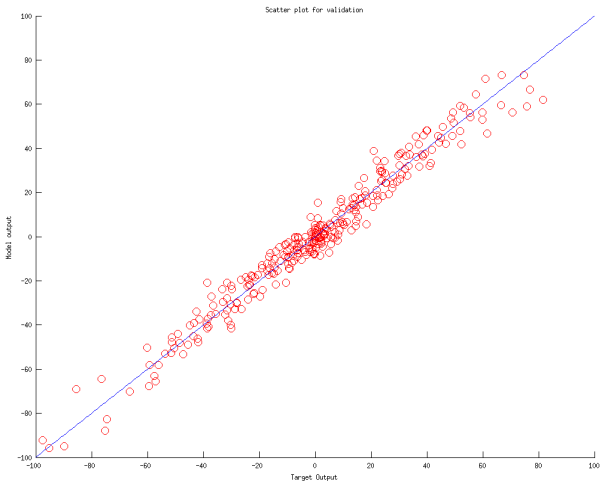
1.2 Dataset 2: 2-D input data

1.2.1 Observation and Inference

- We used 100 data points for training.
- The settings used are ν =0.2, γ=0.015 and C=1000.
- ν-SVR gave a good aproximation of the underlying function.
- Performance of ν-SVR is comparable to RBF and MLFFNN and the surface realisations are almost the same.



(a) Scatter Plot for train



(b) Scatter Plot for validation

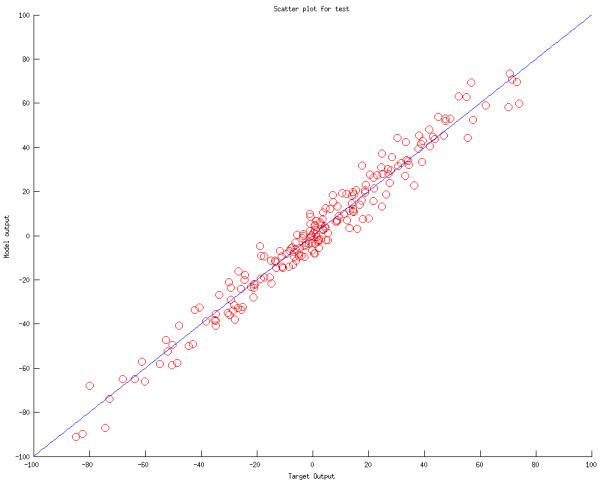
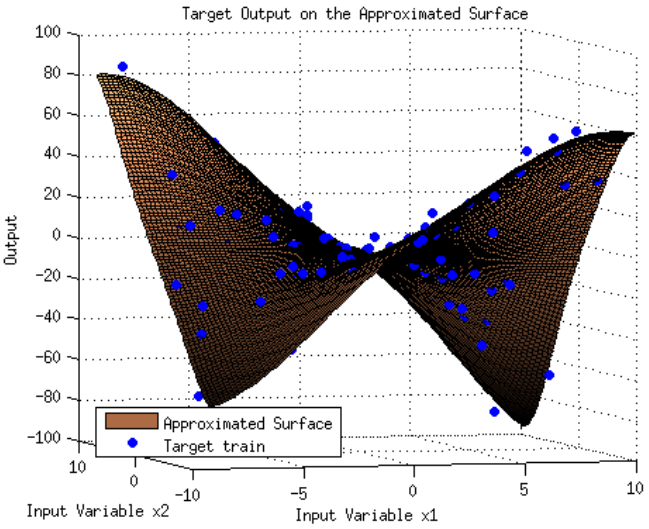
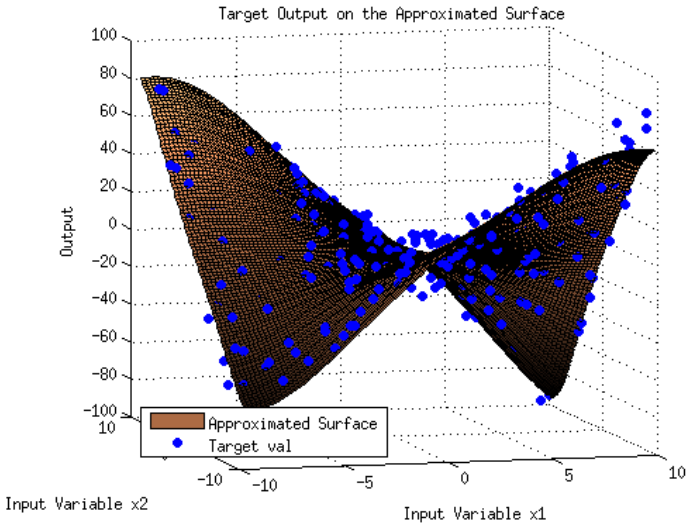


Figure 4: Scatter Plot for test



(a) Approximated Surface vs Target for train



(b) Approximated Surface vs Target for validation

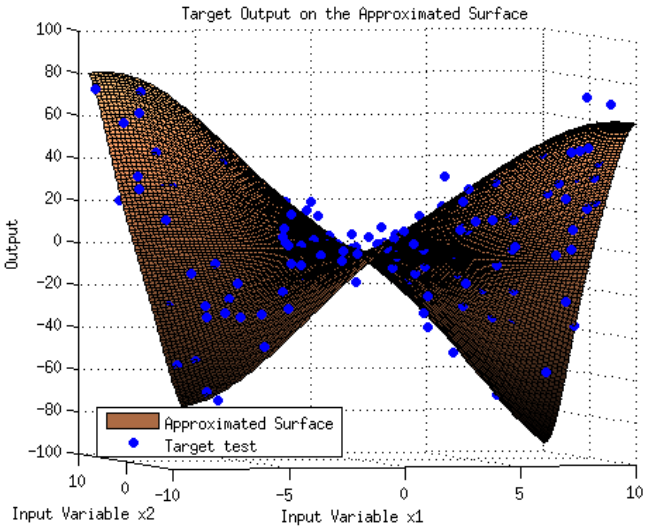


Figure 5: Approximated Surface vs Target for test

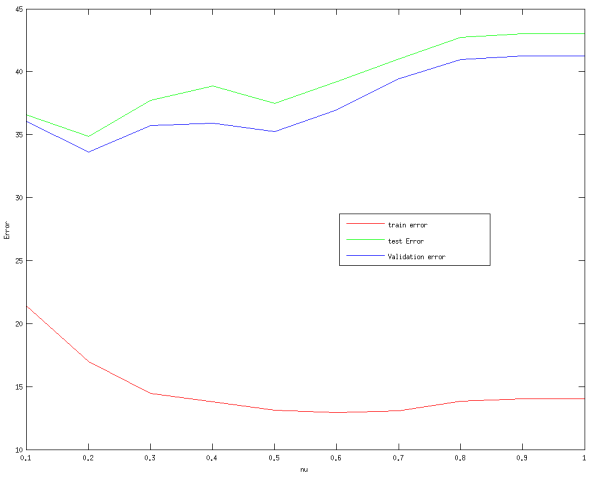


Figure 6: Variation of error vs nu

2 Novelty Detection

2.1 Dataset 3: 2-D Overlapping Classes

2.1.1 Observation and Inference

- The dataset has 4 overlapping classes. Class 1 was taken as the normal class while the rest were abnormal.
- From the confusion matrix we can infer the true postitive rate =0.9 and false alarm rate = 0.0909. The settings used are ν =0.1, γ=0.05.

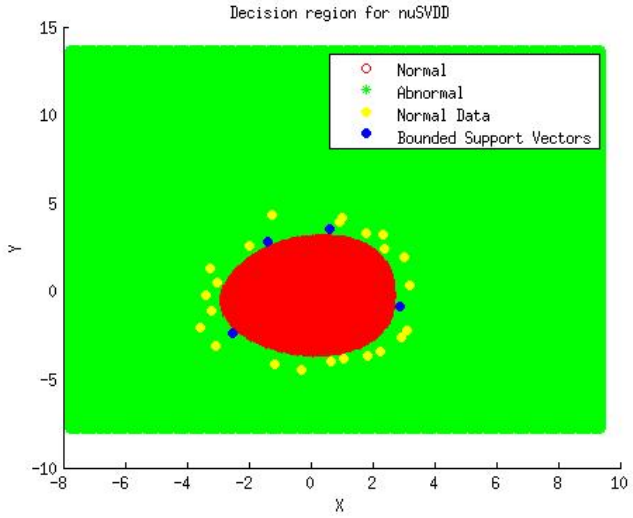


Figure 7: Decision Boundary with Bounded and Unbounded Support vectors

	Abnormal	Normal
Abnormal	291	9
Normal	10	90

Table 1: Confusion matrix for 2-D data

2.2 Dataset 4: Multivariate input data

2.2.1 Observation and Inference

- The dataset has 32 abnormal and 123 normal samples.
- From the confusion matrix we can infer the true postitive rate =0.7826 and false alarm rate = 0.2174. The settings used are ν =0.08, γ=0.001.

	Abnormal	Normal
Abnormal	11	5
Normal	5	18

Table 2: Confusion Matrix for Hepatitis Normal Data

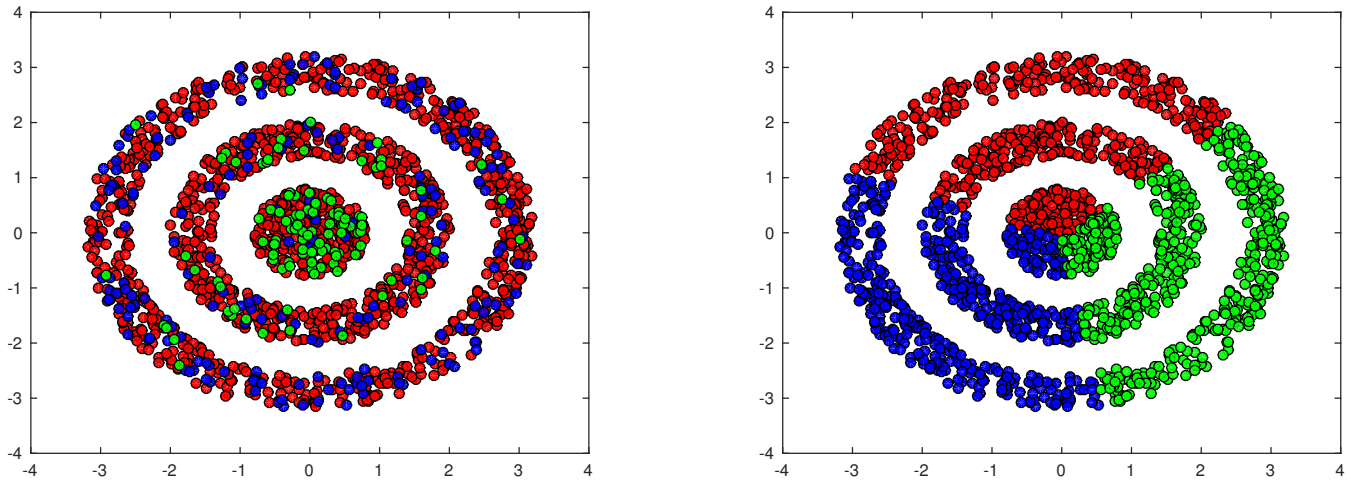
3 Clustering

We are using 2-D non linearly separable data for this task which is having 3 clusters.

3.1 Dataset 5: 2-D Nonlinearly Seperable Classes

3.1.1 Observation and Inference

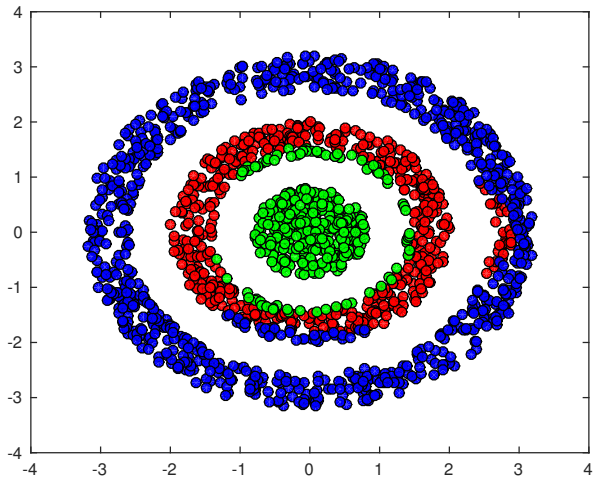
- The initalisation is very crucial in case of kernel k means
- More time consuming since we need to compute the n * n kernal matrix.
- It is able to identify the non-linear structures.



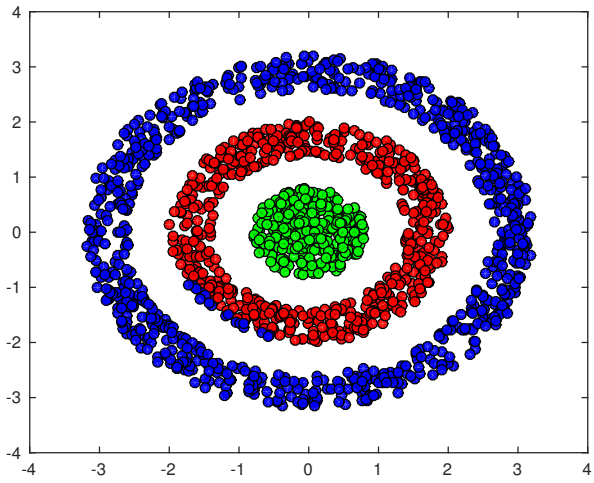
(a) Kernel K-Means initalisation

(b) Normal K-Means Clustering

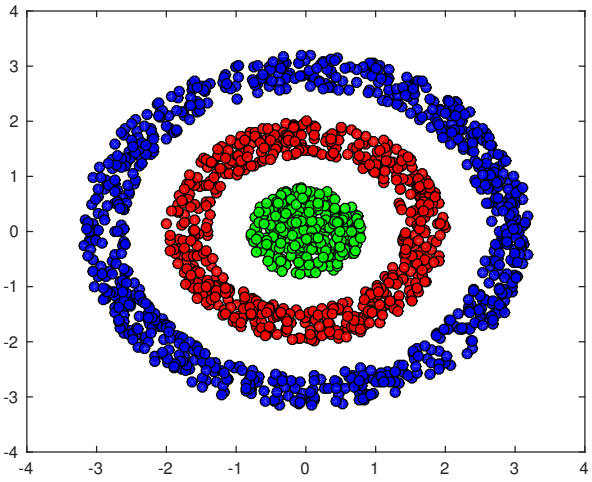
Figure 8: Decision boundaries



(a) After 2nd Iteration

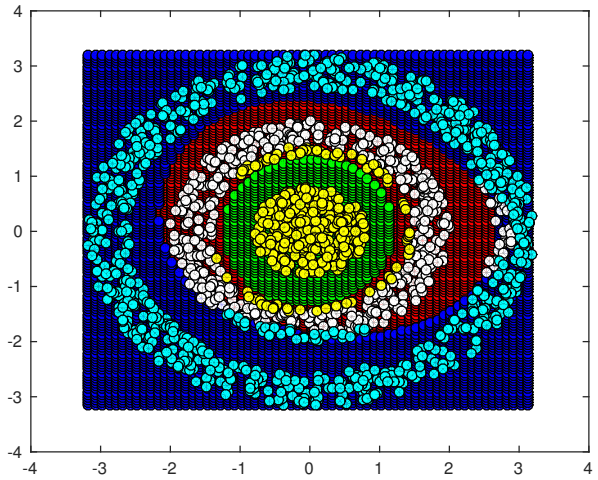


(b) After Intermediate Iteration(4th iteration)

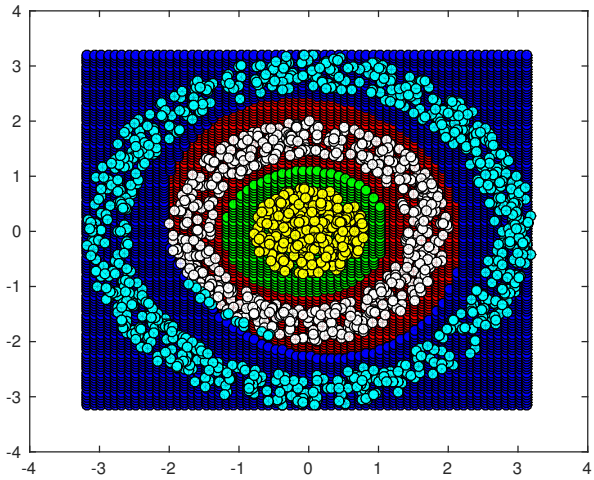


(c) After Convergence

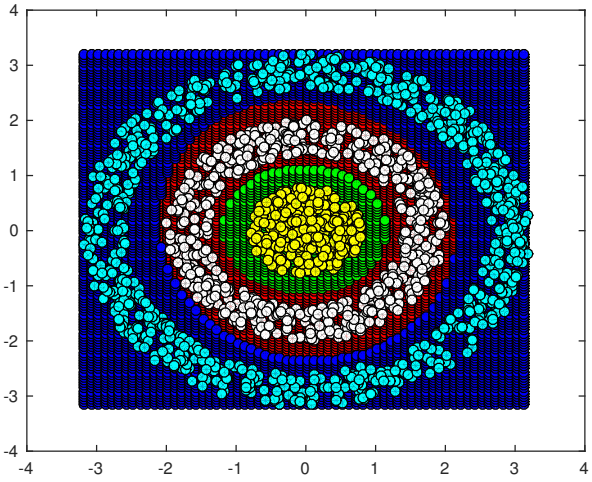
Figure 9: Data points in different clusters



(a) After 2nd Iteration



(b) After Intermediate Iteration(4th iteration)



(c) After Convergence

Figure 10: Decision boundaries for Different iterations

4 Semisupervised Learning

4.1 Dataset 6: 2-D input data (Twomoons)

4.1.1 Observation and Inference

- For two moons data, 2 points were labelled while rest were unlabelled.
- For graph-based semisupervised using label propagation we used 159 unlabelled samples and 39 test samples.
- The g used is 125. We obtained 100% accuracy.
- For ν-SVM $\gamma = 0.06$ and ν=0.06 and 100% accuarcy.
- For self-training ν Svm γ is 0.5 and ν is 0.001
- For S^3VM γ=0.15 and C=10. We saw an accuracy of 54%.
- Among the above methods graph based label propagation gave the best decision region.

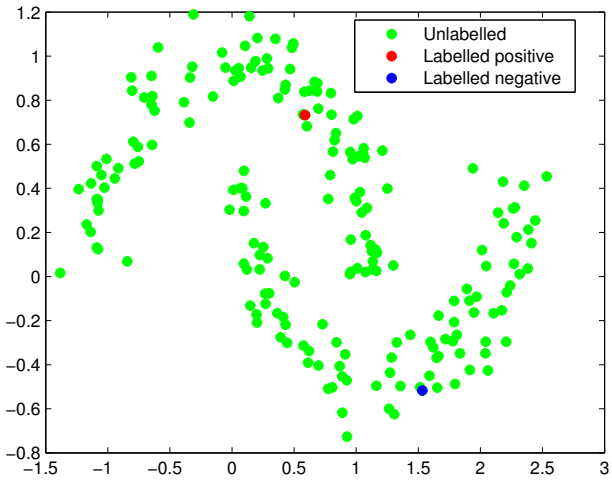
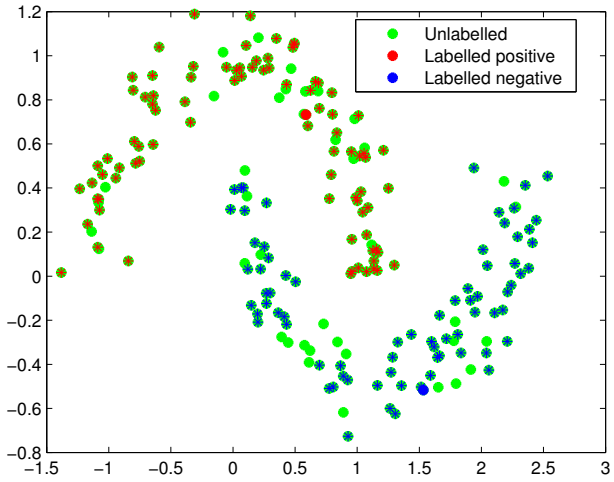


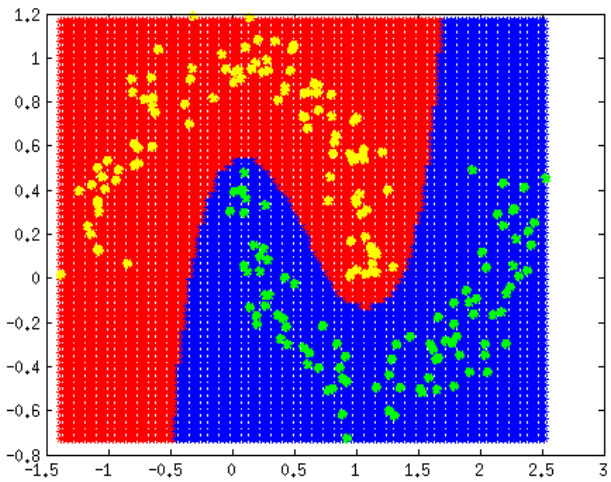
Figure 11: initial scenario

	Class 1	Class 2
Class 1	10	0
Class 2	0	10

Table 3: Confusion Matrix for 2-D data - Supervised ν SVM for gama 0.06 and ν 0.06

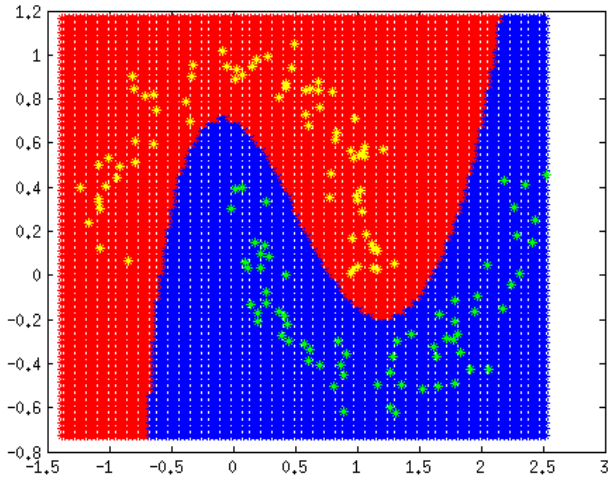


(a) Labelled and Unlabelled examples after propagation

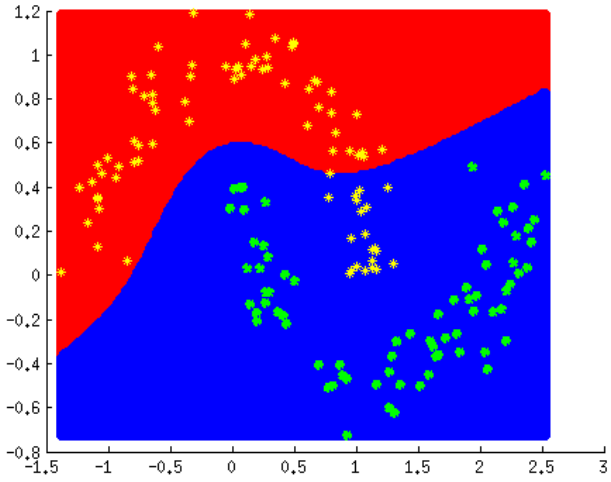


(b) Decision Region

Figure 12: Plots for Graph based semi supervised



(a) Decision Region for supervised nu SVM



(b) Decision Region for Self Training with nu SVM

Figure 13: Decision boundaries

	Class 1	Class 2
Class 1	21	0
Class 2	1	17

Table 4: Confusion Matrix for self training for 39 test samples

Figure 14: Decision boundary for S^3VM

	Class 1	Class 2
Class 1	21	0
Class 2	0	18

Table 5: Confusion Matrix for Graph Based for 39 test samples

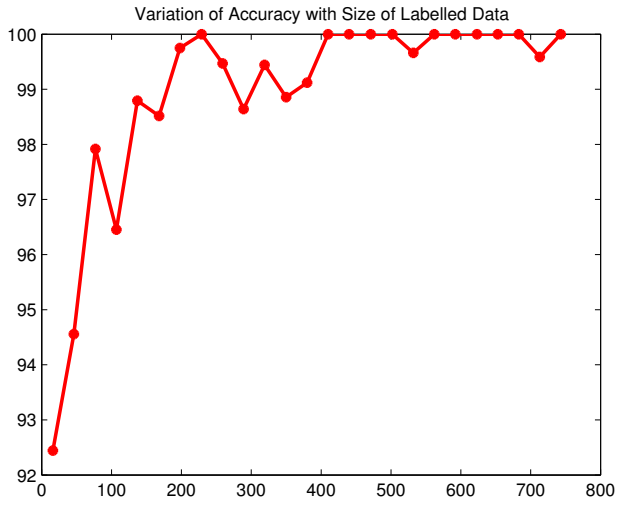
	Class 1	Class 2
Class 1	49	46
Class 2	35	66

Table 6: Confusion Matrix for S^3VM for 150 test samples

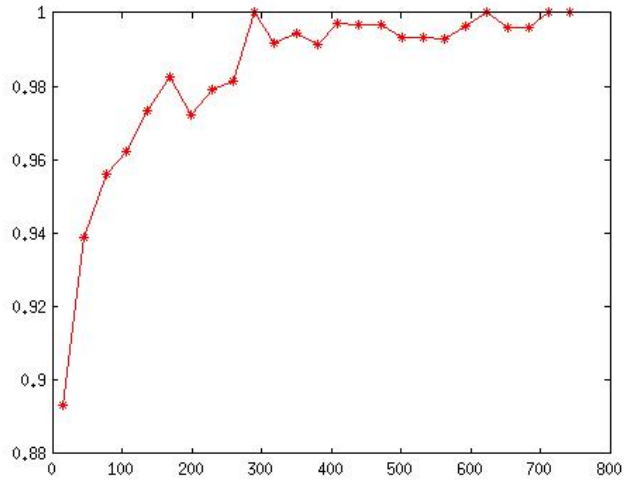
4.2 Dataset 7: UCI dataset

4.2.1 Observation and Inference

- For this data, letters given were I and L. I was taken as positive while L was the negative class.
- The g used is 125. We obtained 100% accuracy.
- For ν-SVM $\gamma = 0.1$ and ν=0.001 and 100% accuarcy. For comparing the variation of accuracy on test data, the fraction of the labelled data were varied from 0.01 to 0.5.
- For the other methods we have used 229 labeled training samples to compare accuracy using confusion matrices.
- For graph-based semisupervised using label propagation the accuracy observed was 97.93%
- For self-training ν Svm γ is 0.01 and ν is 0.001 with 100% accuracy
- For S^3VM γ=0.15 and C=10. We saw an accuracy of 98.67%.



(a) Self-Training with ν SVM



(b) Graph Based Semi Supervised using Label propagation

Figure 15: Variation of Accuracy on Test Data by varying size of labelled Data

	Class 1	Class 2
Class 1	76	0
Class 2	0	76

Table 7: Confusion Matrix for ν SVM

	Class 1	Class 2
Class 1	194	0
Class 2	0	192

Table 8: Confusion Matrix for self Training

	Class 1	Class 2
Class 1	194	0
Class 2	8	184

Table 9: Confusion Matrix for Graph Based

	Class 1	Class 2
Class 1	188	6
Class 2	0	192

Table 10: Confusion Matrix for S^3VM

5 Classification

5.1 Dataset 8: 1-D input data

dataset:

The dataset used is Reuters R8 stemmed data. It contain around 5700 train files and 2800 test files. It contains the classes of documents acq, crude, earn, grain, interest, money-fx, ship, trade.

We choose a subset of this dataset for our experiments. We randomly took 25 examples of each classes for a total of 200 train examples and similarly for test data.

The decaying factor is kept as 0.5 for varying length of substring size to compare the performance of the different approaches.

5.1.1 Inferences

- The N-gram kernel performs the best in almost all cases.
- The N-gram kernel picksup more number of features as we are taking all N sequence substrings.
- The String subseqeunce kernel takes more time as it is dependent on the dynamic programming approach.
- In case of word kernel as it is considering only the words, not the sequence, its preforming poorer compared to the other approaches.

11	2	0	0	4	1	0	7
0	18	0	0	3	0	0	4
1	4	11	0	8	0	0	1
0	1	0	16	2	0	0	6
0	0	0	0	25	0	0	0
0	0	0	0	13	8	0	4
1	5	0	0	3	0	2	14
0	0	0	0	0	0	0	25

Table 11: Confusion Matrix for String Subsequence kernel

25	0	0	0	0	0	0	0
0	24	1	0	0	0	0	0
0	0	25	0	0	0	0	0
0	0	0	25	0	0	0	0
0	0	0	0	19	4	0	2
0	0	0	0	0	24	0	1
0	0	1	0	0	0	24	0
0	0	0	1	0	0	0	23

Table 12: Confusion Matrix for N-gram kernel

Word Kernel	N-gram Kernel	String Subsequence Kernel
62	3 - 28.15%	3- 36%
	4 - 66.83%	4- 58%
	5 - 75.37%	5- 66.5%
	6 - 76.36%	6- 60%
	7 - 76.88%	7- 57%
	8 - 75.37%	8- 54.5%
	9 - 74.37%	
	10 - 70.85%	

Table 13: Confusion Matrix for ν kernel