Assignment 3

Divya Saglani (CS15M041)
Arulkumar S (CS15S023)
Nitish Kumar (CS15S028)

12 May 2016

1 Introduction

In this assignment e are expected to do following tasks :

1. ν-SVR using Guassian Kernel
2. ν-SVDD using Guassian
3. Kernel K-means clustering using guassian kernel
4. Semisupervised Learning
5. Classification of clustered Data

2 Regression

Regression is a task where we are given with the data consisting input and output and we try to approximate the function which can best represent this data.Hence, this is a supervised learning technique where we train the model using training data, select the best model using Cross Validation method and finally can predict the output value of the new data (i.e. test data) using this model.

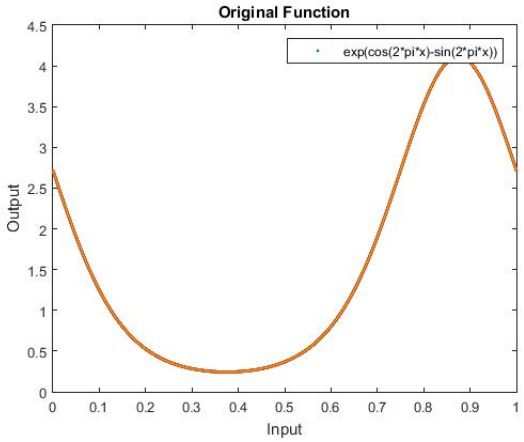
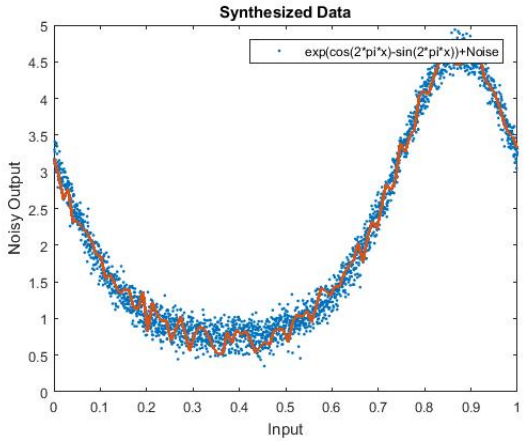
2.1 Univariate data

2.1.1 nu-Support Vector Regression

In this type, we try to approximate the underlying function by tunning the parameters of svmtrain in libsvm.

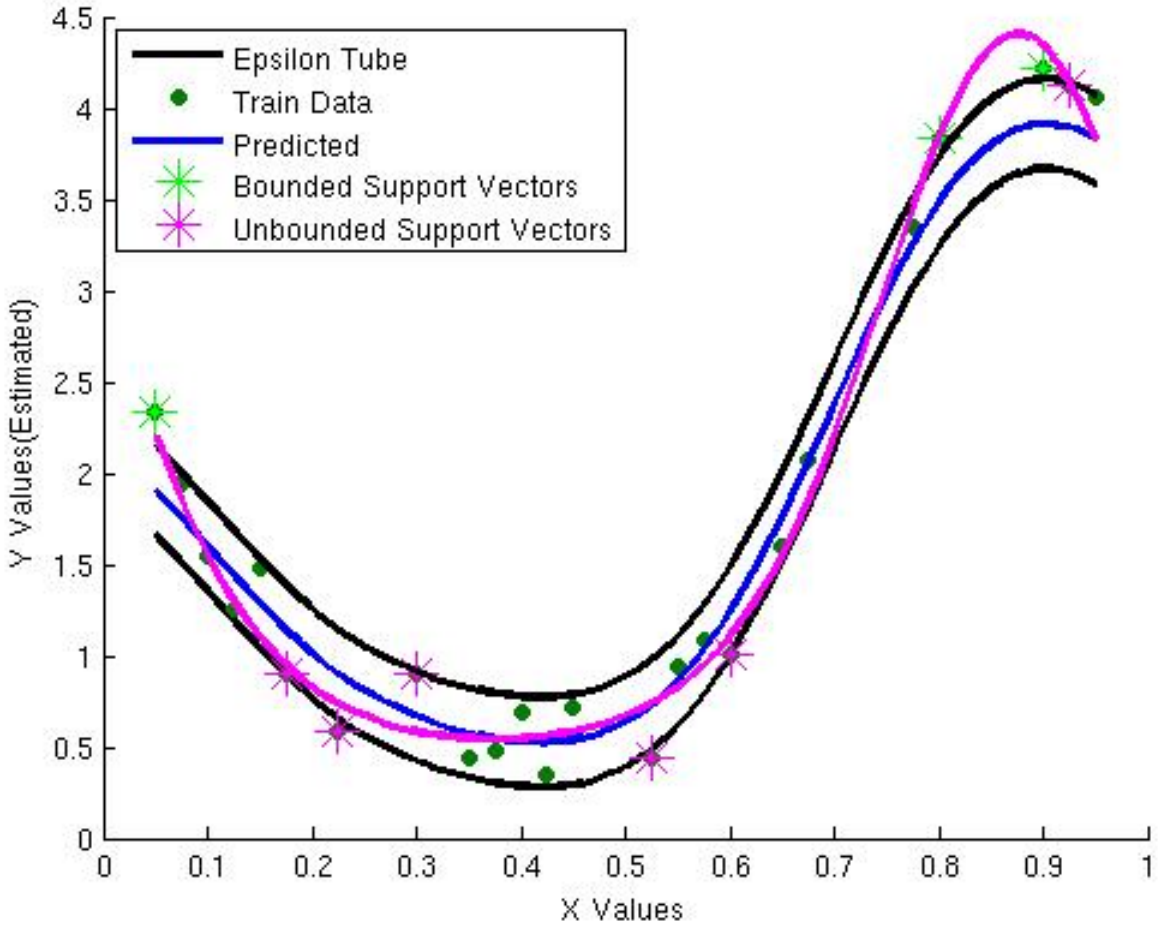
2.1.1.1 Experiments

1. The data was generated by using function $e^{cos(2*\pi*x)-sin(2*\pi*x)}$ and then adding a random noise to it. The random noise was generated using matlab function 'randn' function which generates the numbers with mean zero and variance 1. This noise is then normalised and added to the the actual values.
2. This data was divided in test, train and validation data.The experiments were performed using train data of size 20, validation data of size 8 and test data of size 8.
3. The model was chosen by cross-validation method

Original Data	Synthesized Data
	

4. The best approximation Function we got for the following parameters :
   γ (g) : 16.5
   cost (c) : 1
   ν (n) : 0.3

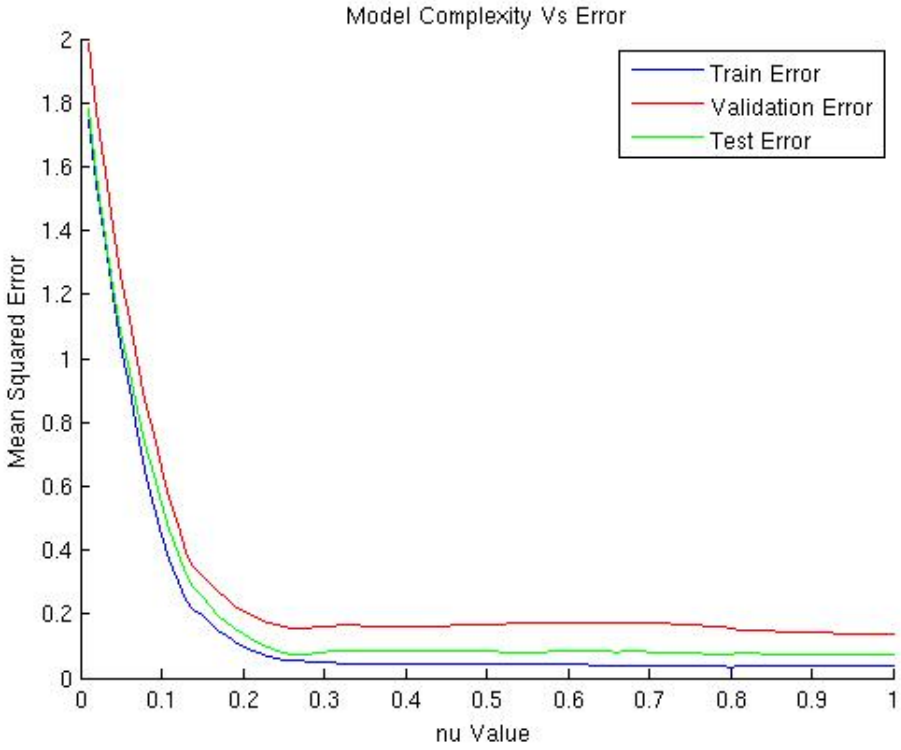
Approximated Function and ϵ Tube



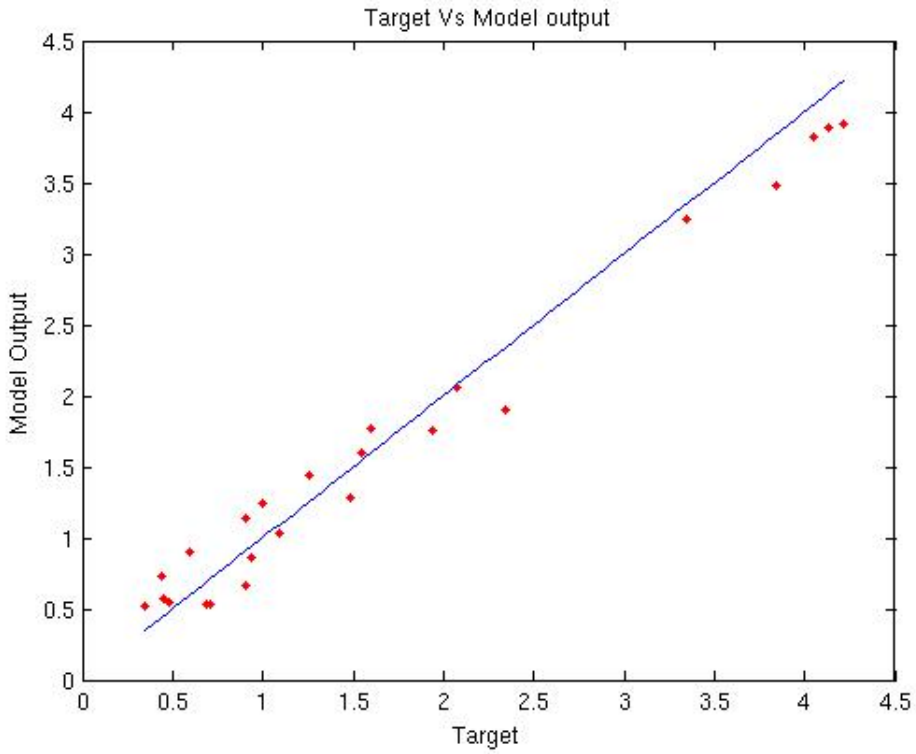
5. Number of bounded support vectors : 3 Number of unbounded support vectors : 5 ϵ value : 0.2392

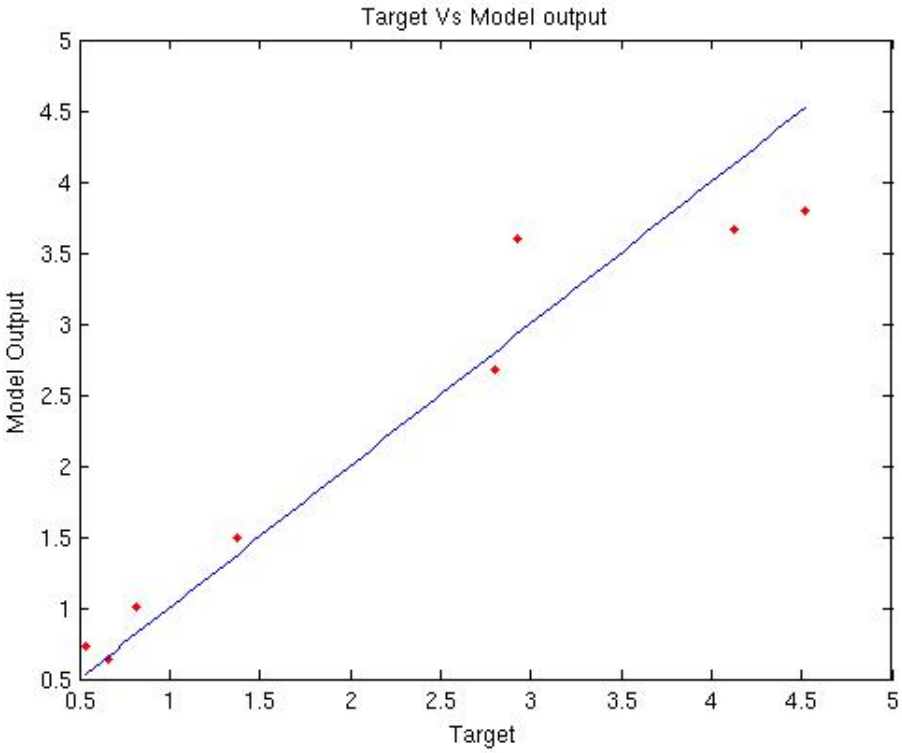
6. Mean squared error for Train Data = 0.0472877
   Mean squared error for Validation Data = 0.0800588
   Mean squared error for Test Data = 0.158917

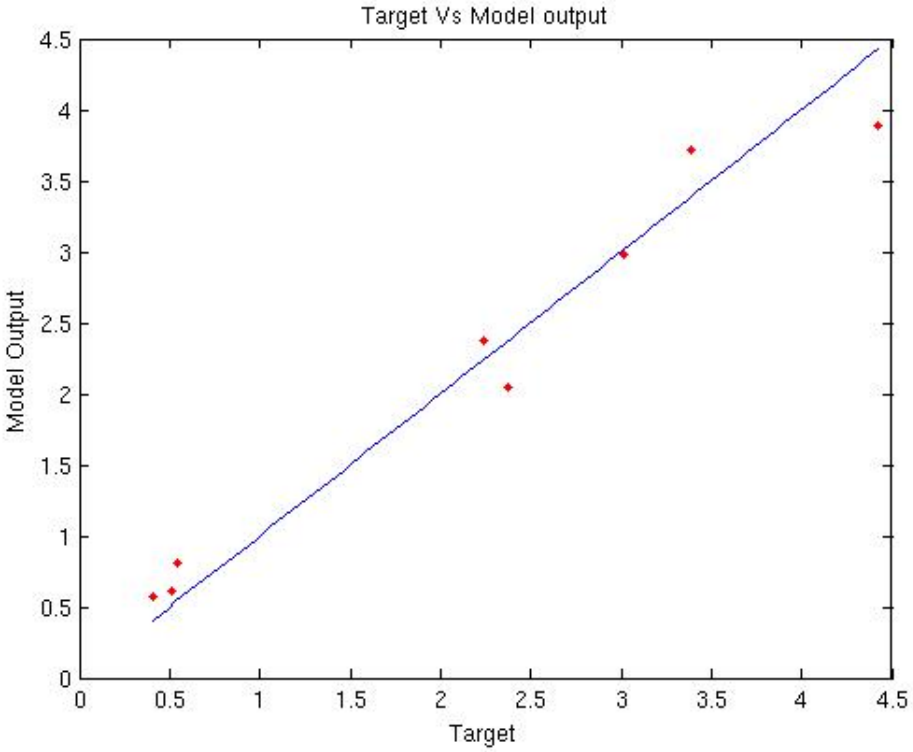
7. The effect of ν value on the Mean Squared error of the approximated function was compared.



8. The scatter plots for test, train and validation data on Actual output Vs predicted output is :




Target Vs Model output
Model Output
Target
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Target Vs Model output
Model Output
Target
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

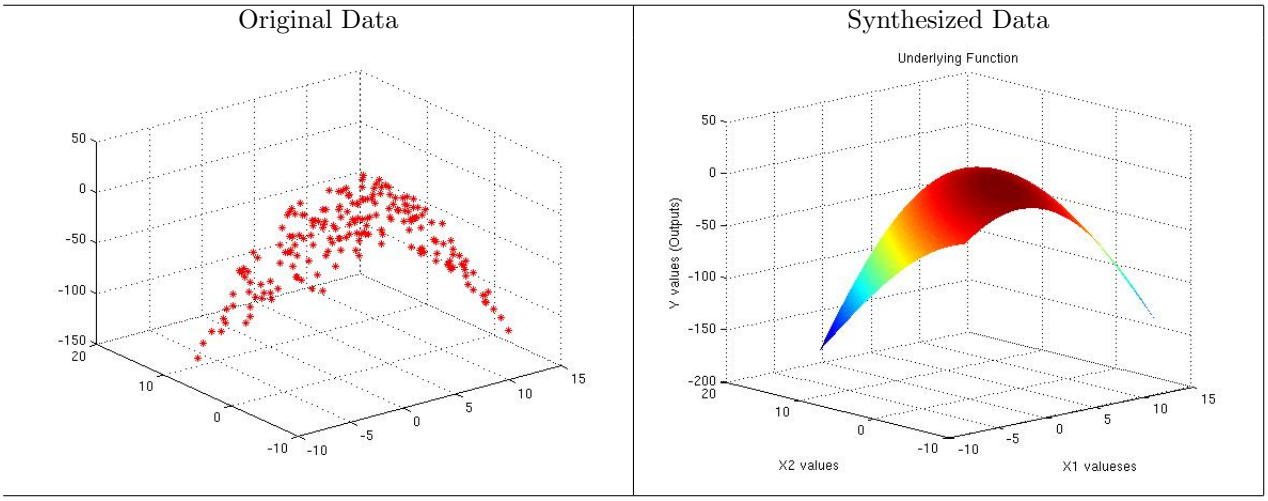
2.2 Bivariate data

2.2.1 nu-Support Vector Regression

In this type, we try to approximate the underlying function by tunning the parameters of svmtrain in libsvm.

2.2.1.1 Experiments

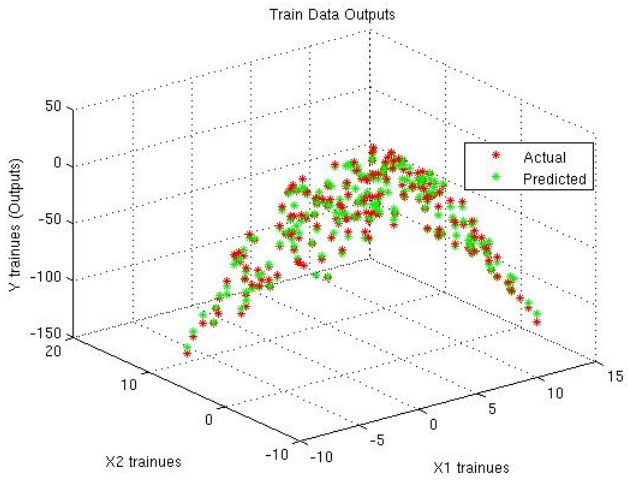
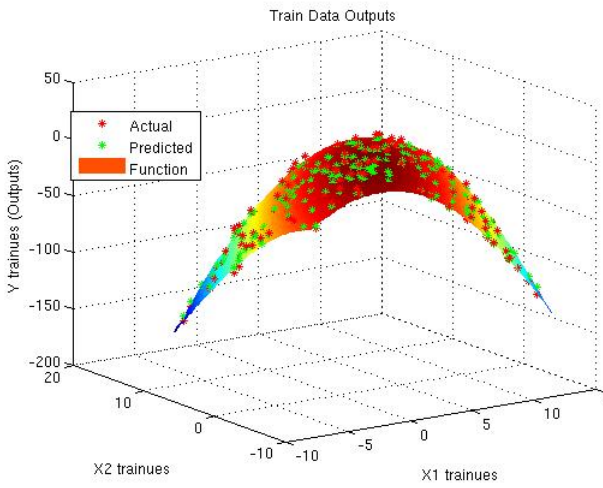
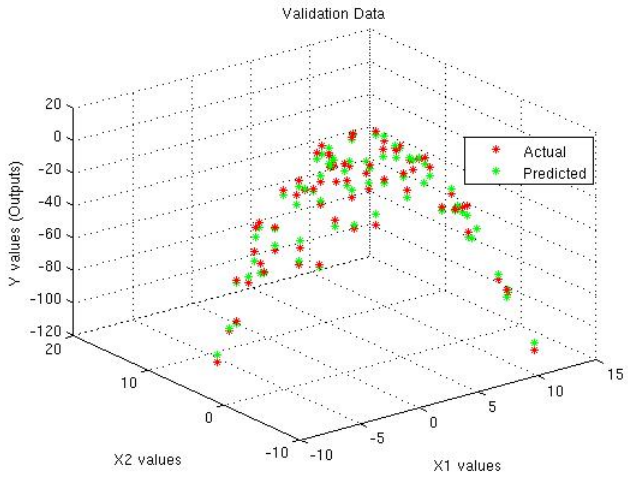
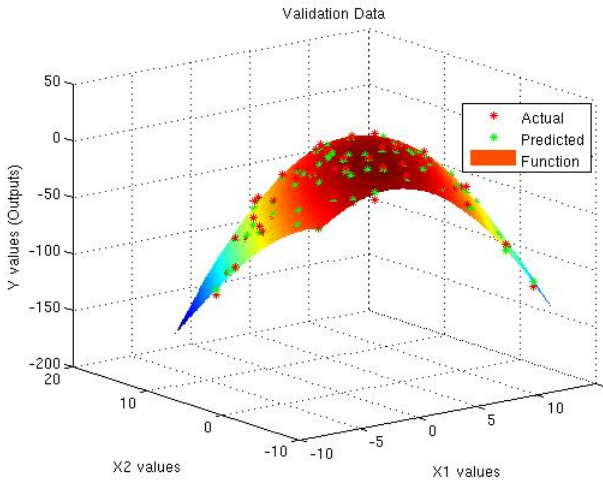
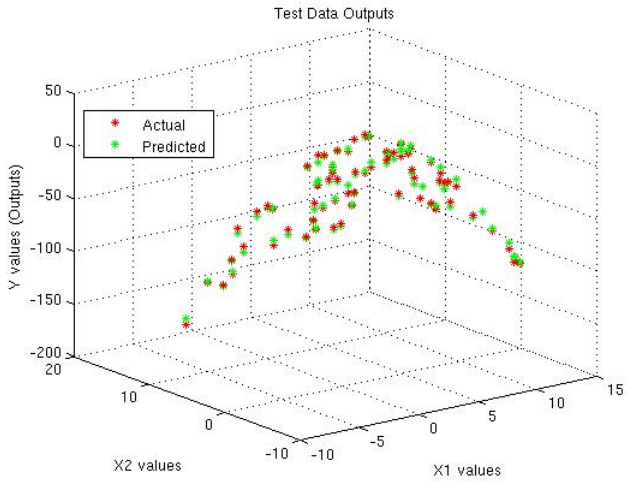
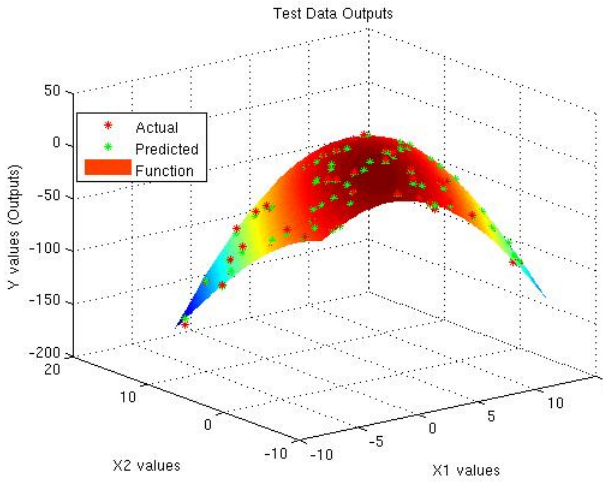
1. This given to dimension data was divided in test, train and validation data.The experiments were performed using train data of size 200, validation data of size 70 and test data of size 70.



2.

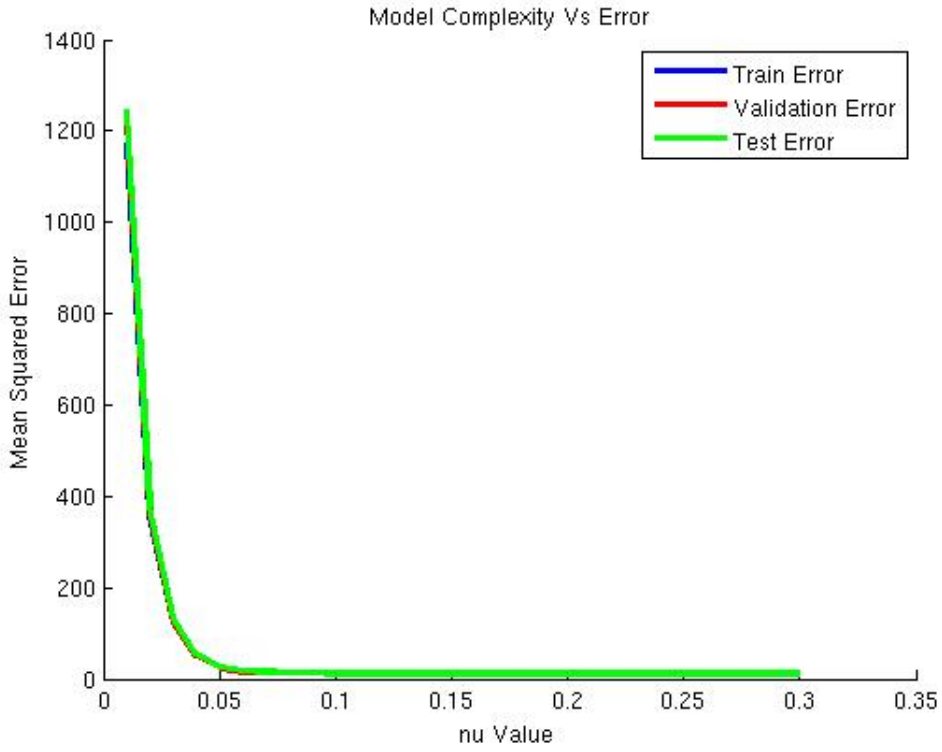
The model was chosen by cross-validation method

The best approximation Function we got for the following parameters :
γ (g) : 0.00095
cost (c) : 700
ν (n) : 0.3

Outputs	Approximated Function
	
	
	

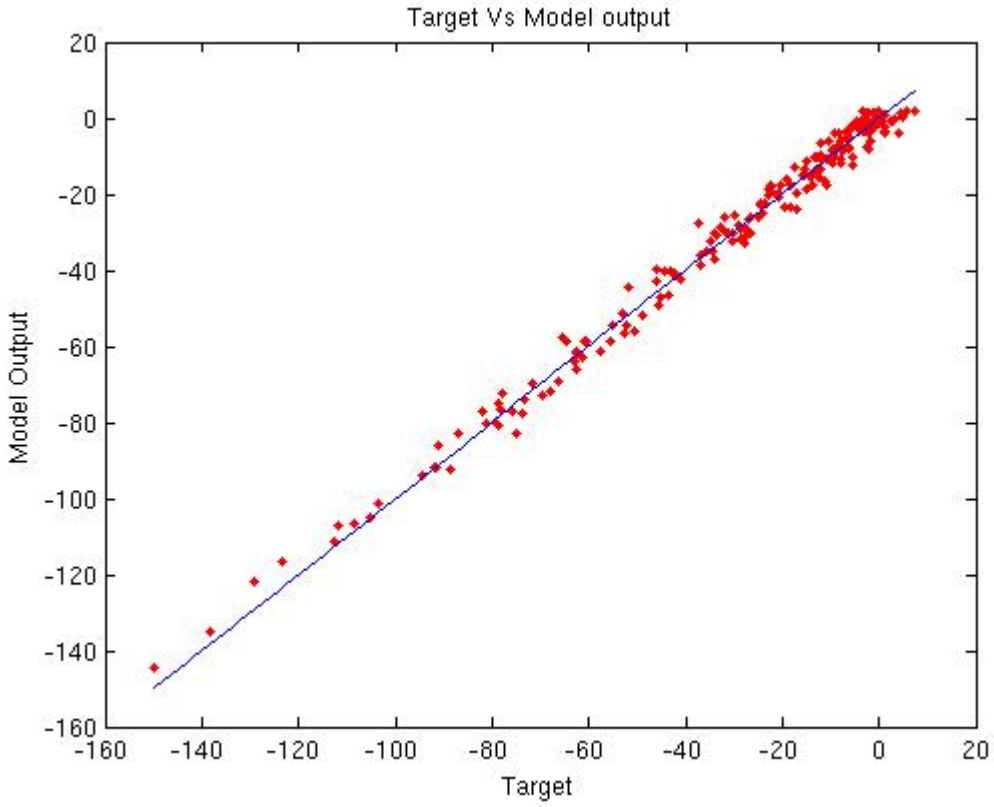
3. Mean squared error = 10.8397
   Mean squared error = 11.3734
   Mean squared error for Test Data = 11.3531

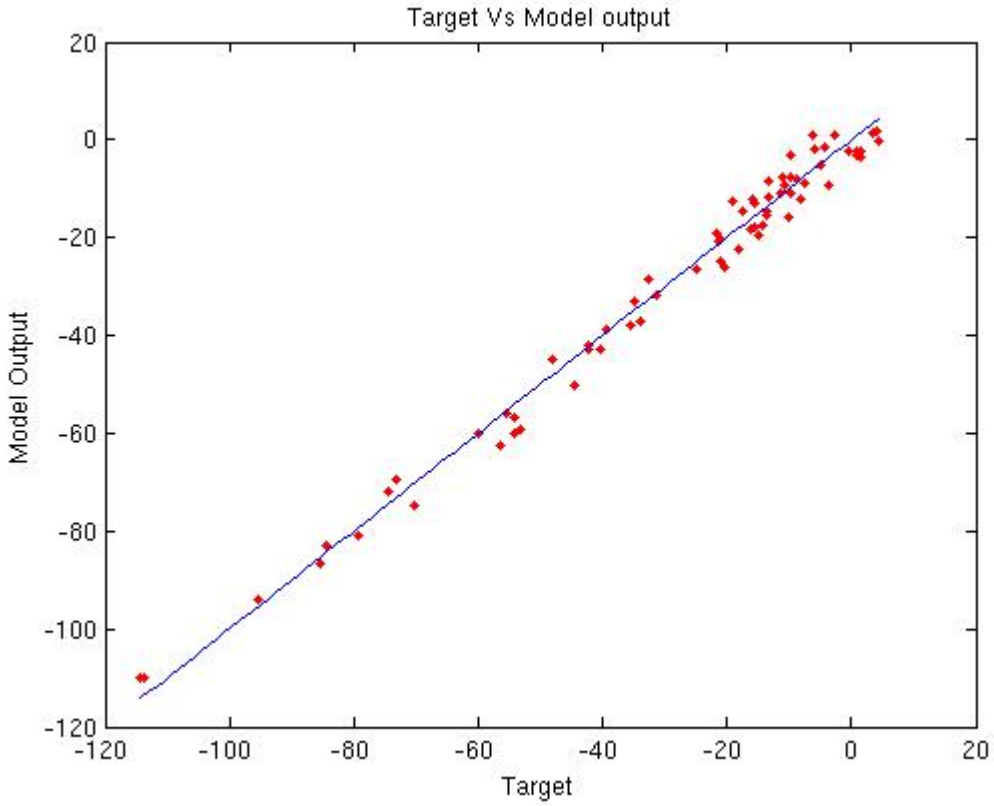
4. The effect of ν value on the Mean Squared error of the approximated function was compared.

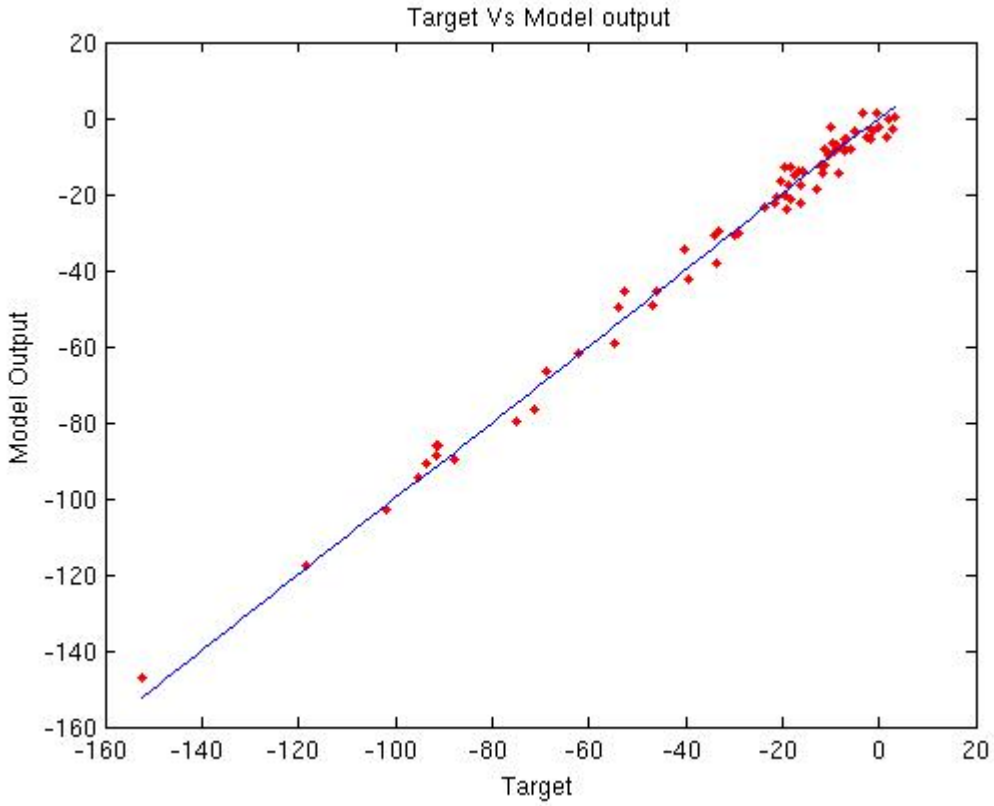


5. The Model Output, Target Output and Approximated function plots for train, test and validation data are:

   The scatter plots for test, train and validation data on Actual output Vs predicted output is :




Target Vs Model output
Model Output
Target
20
0
-20
-40
-60
-80
-100
-120
-120
-100
-80
-60
-40
-20
0
20


Target Vs Model output
Model Output
Target
20
0
-20
-40
-60
-80
-100
-120
-140
-160
-160
-140
-120
-100
-80
-60
-40
-20
0
20

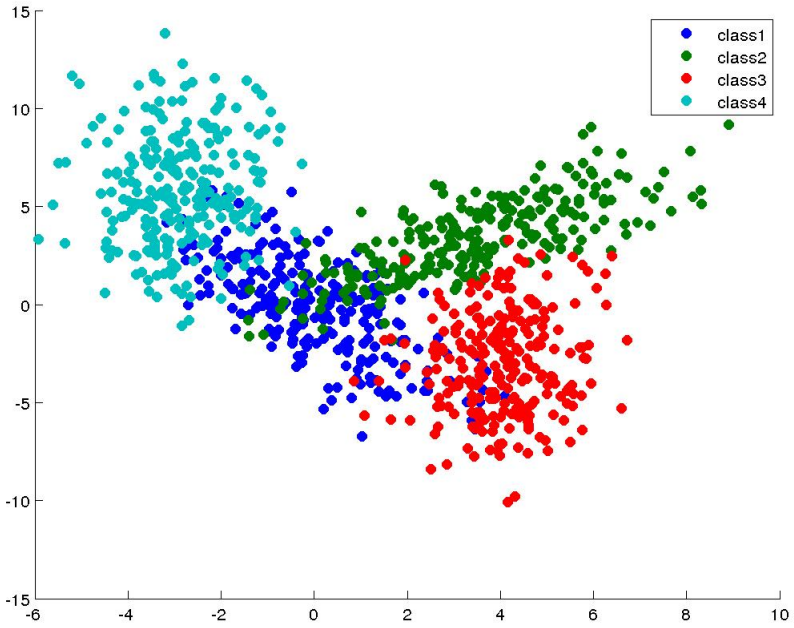
3 Inferences

1. As we increase ν value, the mean squared error decreases.
2. As we increase the γ value the model gets tunned to the training data and hence leads to overfitting.
3. nu-SVR performs better than MLFNN, RBF and linear model for regression . In fact, nu-SVR's performance is comparable with MLFNN but it definitely performs better than RBF and Linear model for Regression for univariate data.
4. For bivariate data all the model outputs were equally good.

3 Support Vector Data Description (SVDD) (one class SVM)

3.1 Dataset - 3 (2-dimensional overlapping data)

3.1.1 Data



The class-1 data (plotted in Blue color) from the dataset-3 is selected as 'normal data' to be learned and represented by one-class SVM.

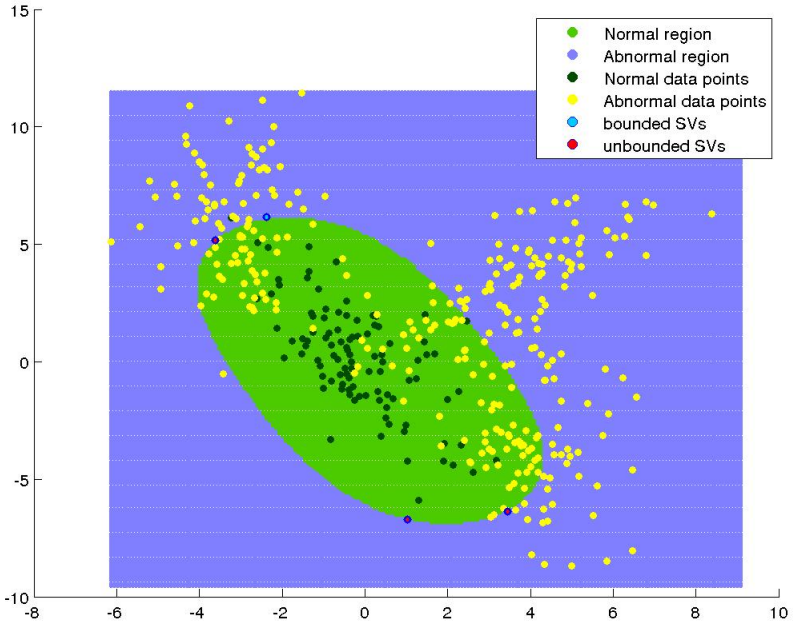
3.1.2 Procedure

- From the 2-dimensional overlapping data, one class (Class-1 with 250 data points) is selected as the 'normal' class used for training one class SVM.
- The data from other classes are uniformly sampled to form the validation (600 data points) and test set (400 data points), which are used to verify the efficiency of the trained model.
- The one-class SVM (with ν as hyperparameter) classifier is trained using different ν values and the best model is choosen based on the performance in validation set.

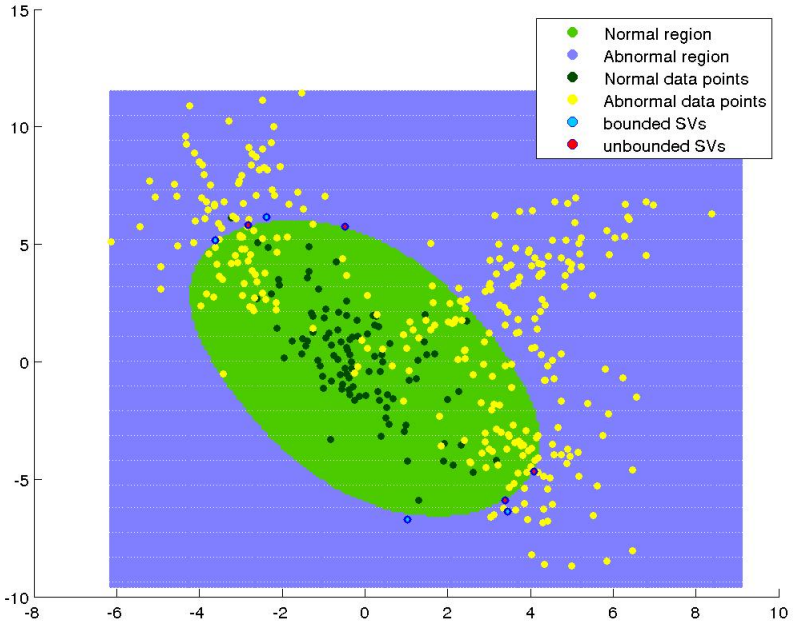
3.1.3 Decision regions

3.1.3.1 Illustration: Almost-Hard-minimal hypersphere SVDD

when $\nu = 0.01$, the SVDD model allows very less outliers. (i.e., Bounded support vectors = 1% of the training data).
Hence, the model trained by having ν as 0.01 has the disadvantage of classifying all other class data which are included inside the hypersphere, as the 'Normal' class. This behaviour will affect the performance of classifier, as shown in the figure 1.



(a) One-class SVM with RBF kernel (gamma = 0.01, $\nu = 0.01$, bounded SVs = 1, unbounded SVs = 3)



(b) One-class SVM with RBF kernel (gamma = 0.01, $\nu = 0.02$, bounded SVs = 4, unbounded SVs = 4)

Figure 1: illustration of Almost-hard-minimal hypersphere which gives poor performance

3.1.3.2 Confusion matrices & performance details

3.1.3.2.1 $\nu = 0.01$, RBF kernel parameter (γ) = 0.01

Validation data

	Abnormal class (predicted)	Normal class (predicted)
Abnormal class (Target)	289	161
Normal class (Target)	4	146

	Formula	Value
True positive rate w.r.t., Normal class (recall)	$\frac{TP}{TP+FN}$	0.973
False positive rate w.r.t., Normal class (fall-out)	$\frac{FP}{TN+FP}$	0.357
F1-score	$\frac{2TP}{2TP+FN+FP}$	0.638
Accuracy		68.75%

Test data

	Abnormal class (predicted)	Normal class (predicted)
Abnormal class (Target)	176	124
Normal class (Target)	1	99

	Formula	Value
True positive rate w.r.t., Normal class (recall)	$\frac{TP}{TP+FN}$	0.999
False positive rate w.r.t., Normal class (fall-out)	$\frac{FP}{TN+FP}$	0.413
F1-score	$\frac{2TP}{2TP+FN+FP}$	0.613
Accuracy		68.75%

3.1.3.2.2 $\nu = 0.02$, RBF kernel parameter (γ) = 0.01

Validation data

	Abnormal class (predicted)	Normal class (predicted)
Abnormal class (Target)	285	163
Normal class (Target)	5	145

	Formula	Value
True positive rate w.r.t., Normal class (recall)	$\frac{TP}{TP+FN}$	0.966
False positive rate w.r.t., Normal class (fall-out)	$\frac{FP}{TN+FP}$	0.362

F1-score	$\frac{2TP}{2TP+FN+FP}$	0.632
Accuracy		68.5%

Test data

	Abnormal class (predicted)	Normal class (predicted)
Abnormal class (Target)	175	125
Normal class (Target)	1	99

	Formula	Value
True positive rate w.r.t., Normal class (recall)	$\frac{TP}{TP+FN}$	0.99
False positive rate w.r.t., Normal class (fall-out)	$\frac{FP}{TN+FP}$	0.417
F1-score	$\frac{2TP}{2TP+FN+FP}$	0.611
Accuracy		68.75%

3.1.3.3 Soft-minimal hypersphere for better performance

To get an optimal soft-minimal hypersphere, ν can be set as 0.1 or 0.2 to allow 10% or 20% outliers (Bounded support vectors) of the data to lie out of the minimal hypersphere. This will give higher performance, as it will reduce the misclassification rate, as shown in the figure 2.

3.1.3.3.1 $\nu = 0.25$, RBF kernel parameter $(\gamma) = 0.01$

Validation data

	Abnormal class (predicted)	Normal class (predicted)
Abnormal class (Target)	408	42
Normal class (Target)	31	119

	Formula	Value
True positive rate w.r.t., Normal class (recall)	$\frac{TP}{TP+FN}$	0.793
False positive rate w.r.t., Normal class (fall-out)	$\frac{FP}{TN+FP}$	0.0933
F1-score	$\frac{2TP}{2TP+FN+FP}$	0.765
Accuracy		87.83%

Test data

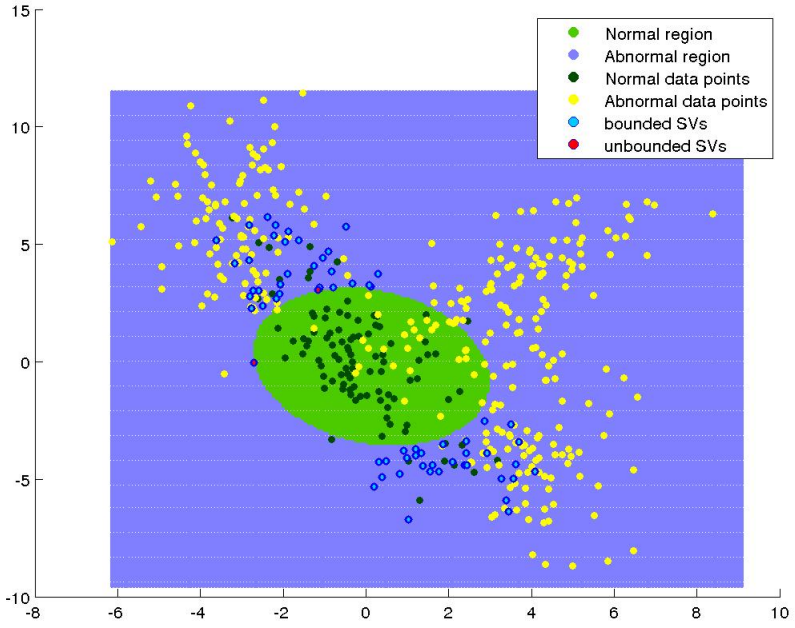


Figure 2: illustration of optimal soft-minimal hypersphere which gives good performance. One-class SVM with RBF kernel (gamma = 0.01, $\nu = 0.25$, bounded SVs = 61, unbounded SVs = 2)

	Abnormal class (predicted)	Normal class (predicted)
Abnormal class (Target)	269	31
Normal class (Target)	22	78

	Formula	Value
True positive rate w.r.t., Normal class (recall)	$\frac{TP}{TP+FN}$	0.78
False positive rate w.r.t., Normal class (fall-out)	$\frac{FP}{TN+FP}$	0.103
F1-score	$\frac{2TP}{2TP+FN+FP}$	0.746
Accuracy		86.75%

3.2 Dataset-4 : Breast benign

3.2.1 Procedure

- The given normal data (458 data points) is splitted into 70% as training set, 10% as validation set, the remaining 20% as test set.
- The given abnormal data (241 data points) is split into 50% as validation set and 50% as test set.
- One-class SVM (with RBF kernel) is trained only with Normal Data and the hyper-parameter(ν) is fixed using validation set.
- The efficiency of chosen model is evaluated using test set and results are reported.

The best validation performance of 94.54% is obtained, when RBF kernel parameter (γ) = 0.001 and ν = 0.05, with Bounded SVs = 13, unbounded SVs = 4.

3.3 Confusion matrices and performance details

3.3.1 Validation data

	Abnormal class (predicted)	Normal class (predicted)
Abnormal class (Target)	116	4
Normal class (Target)	5	40

	Formula	Value
True positive rate w.r.t., Normal class (recall)	$\frac{TP}{TP+FN}$	0.889
False positive rate w.r.t., Normal class (fall-out)	$\frac{FP}{TN+FP}$	0.0334
F1-score	$\frac{2TP}{2TP+FN+FP}$	0.898
Accuracy		94.54%

3.3.2 Test data

	Abnormal class (predicted)	Normal class (predicted)
Abnormal class (Target)	118	3
Normal class (Target)	4	89

	Formula	Value
True positive rate w.r.t., Normal class (recall)	$\frac{TP}{TP+FN}$	0.956
False positive rate w.r.t., Normal class (fall-out)	$\frac{FP}{TN+FP}$	0.025
F1-score	$\frac{2TP}{2TP+FN+FP}$	0.962
Accuracy		96.72%

4 Clustering

4.1 K-Means

K-Means is unsupervised learning algorithm.Data is highly non-linearly distributed.Shape of Data is divided into three circle for three class.our Data is not linearly separable. We used k is 3. where k is number of cluster.we initialize the centre randomly within the data. Then we find distance for each data points for each centre.From each centre points is minimum distance we assign to that cluster.Convergence criteria is when no more points change previous iteration to current iteration.Data is converged in 6 iteration.

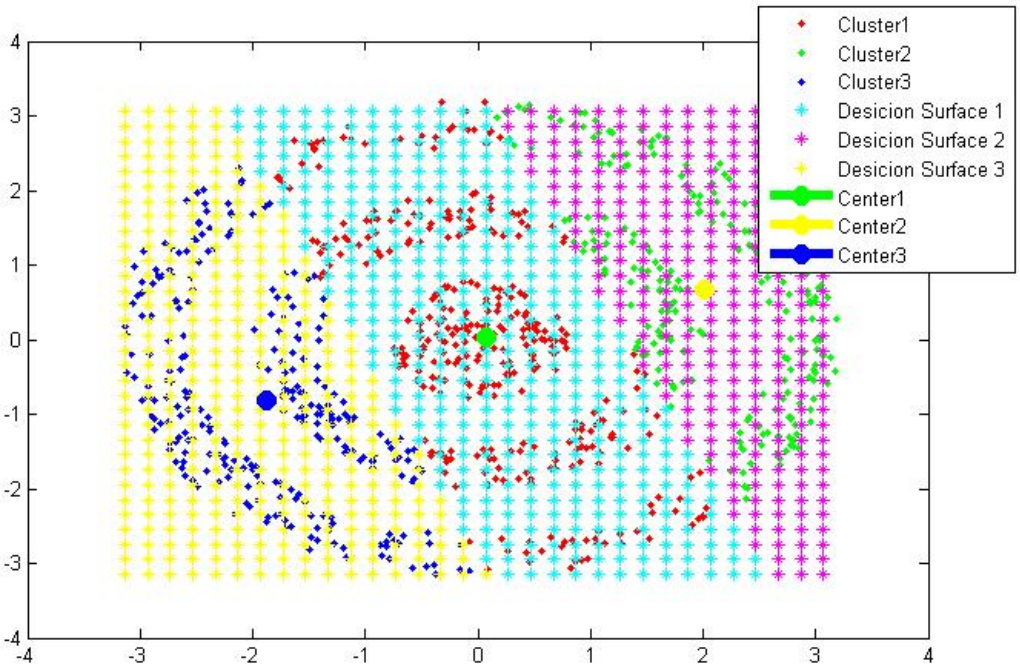


Figure 1: After Initialization

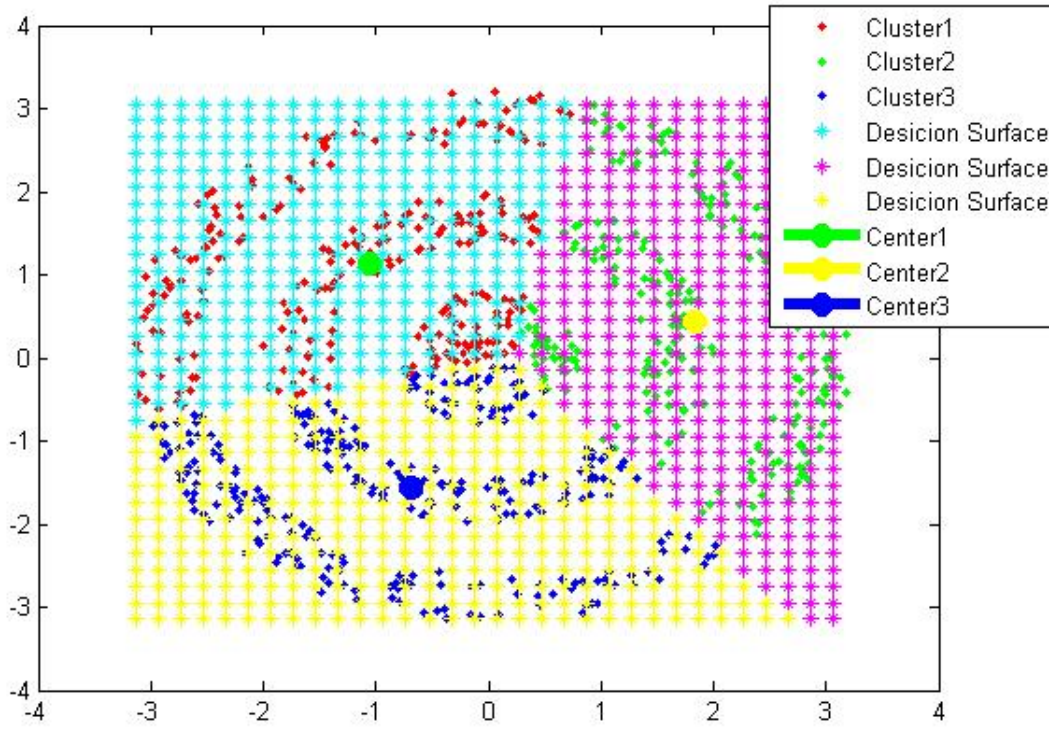


Figure 2: Intermediate

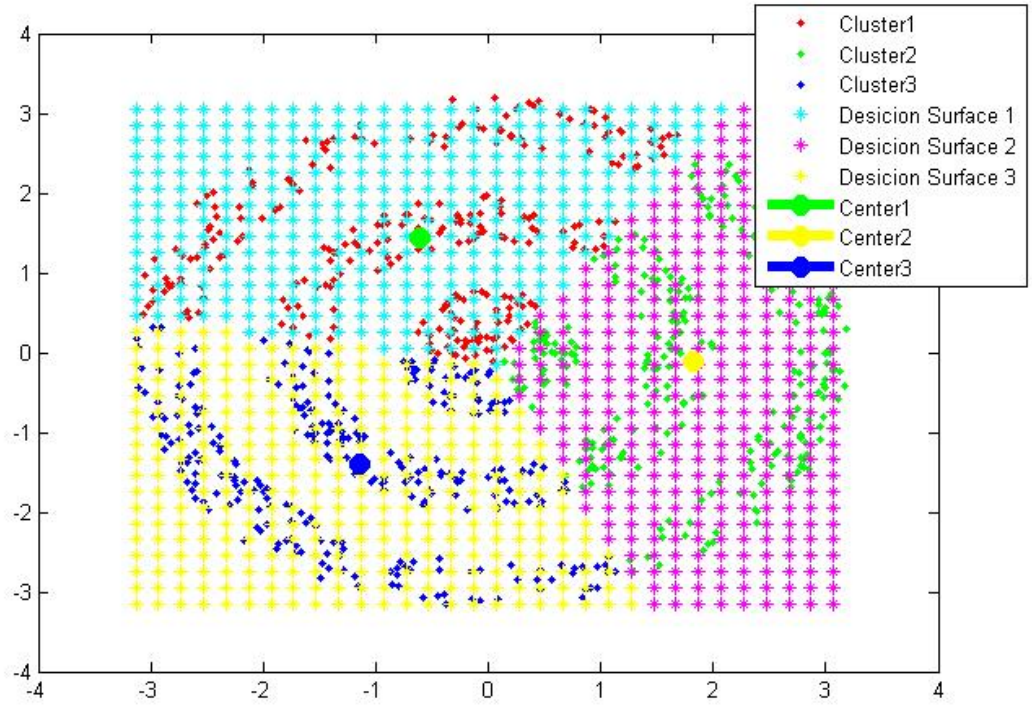


Figure 3: After convergence

4.2 Kernel K-Means

Here data is not linearly separable.So we use kernel function that transform data into other space.We use Gaussian kernel.Data is first randomly initialize to three different cluster.For Gaussian kernel we take Gamma value is 0.74.It converged in 10 iteration.

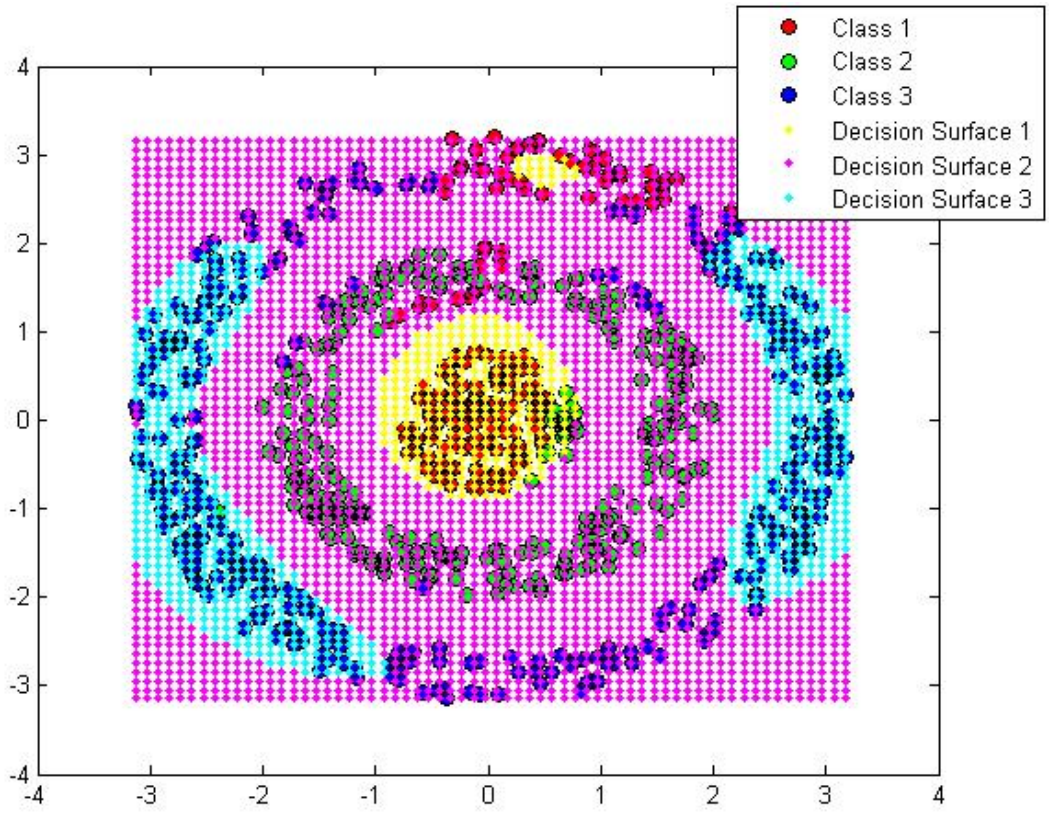


Figure 4: After Initialization

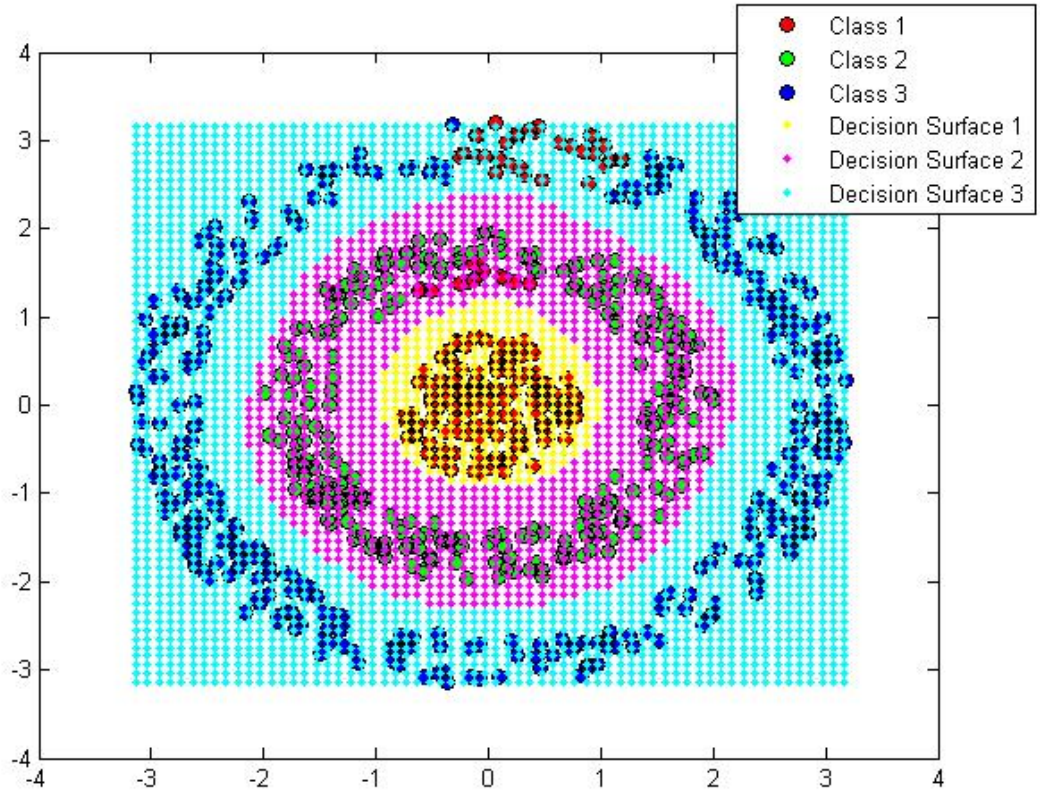


Figure 5: After Second iteration

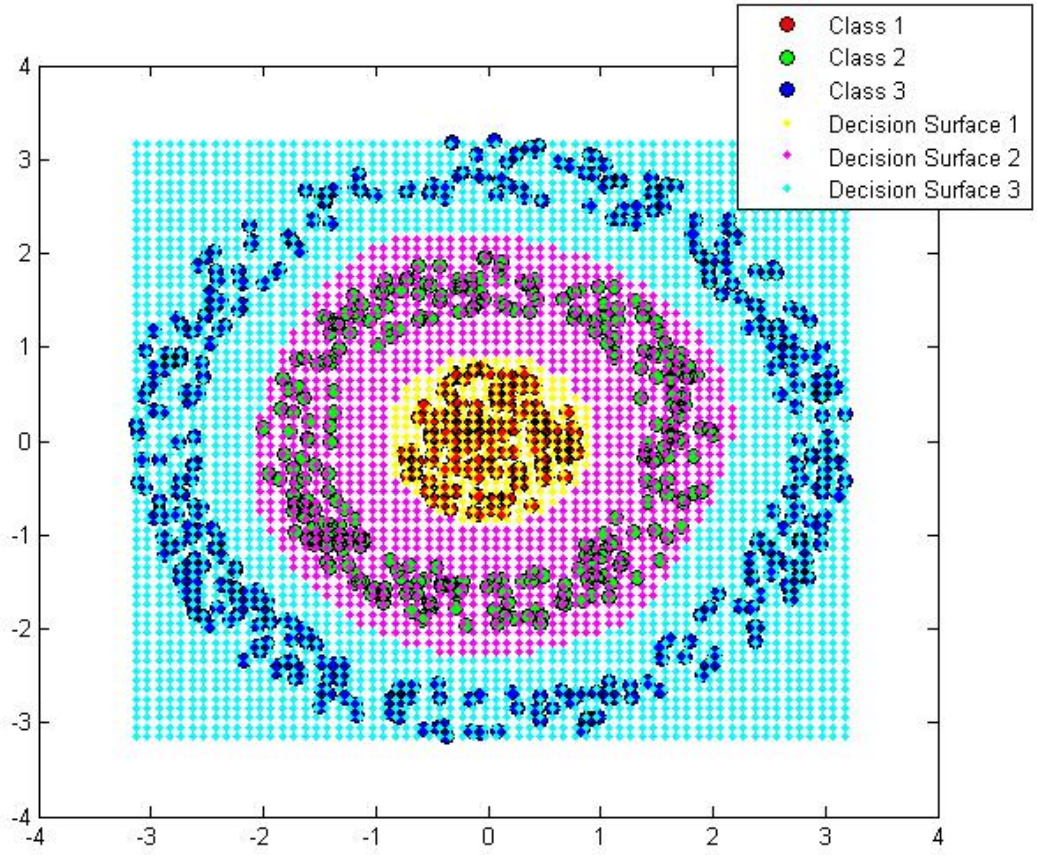


Figure 6: Intermediate

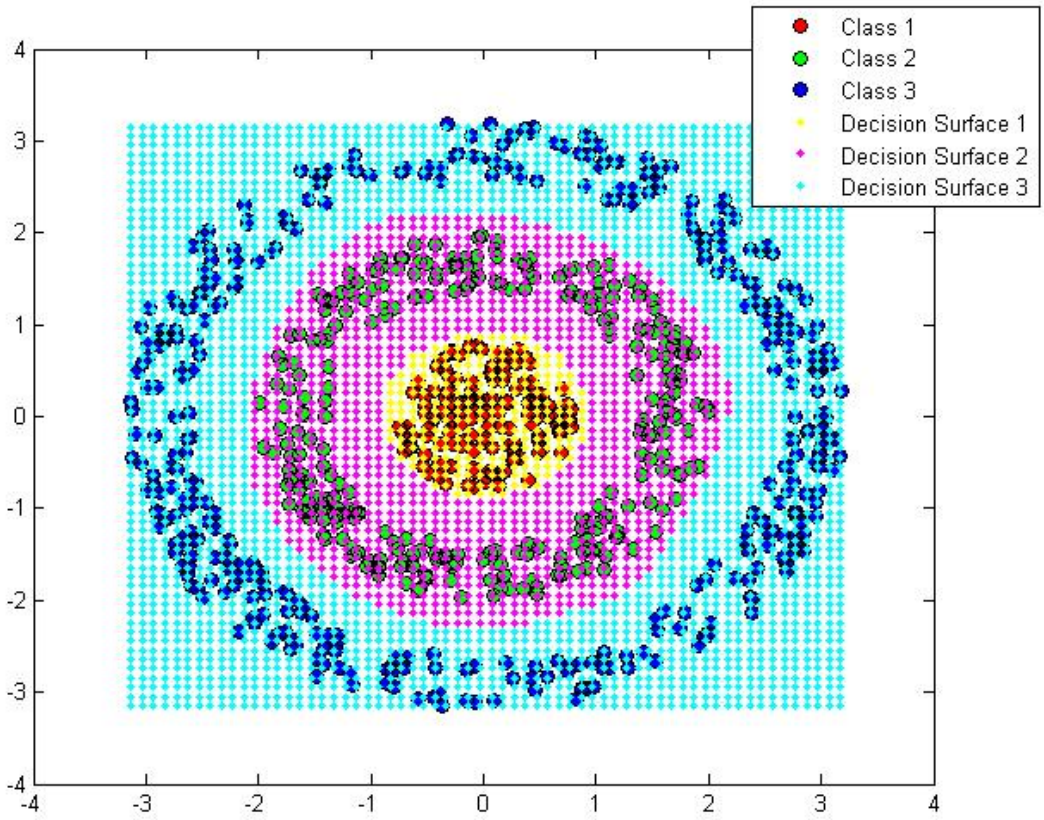


Figure 7: After convergence

5 Semi Supervised Learning

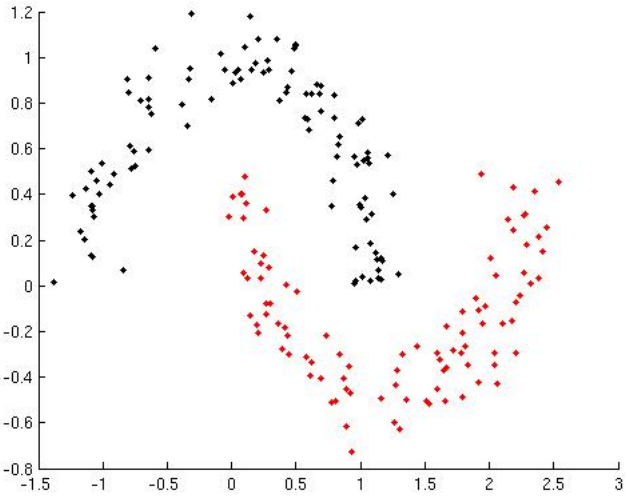
Semi-Supervised data is a mixture of labelled and unlabelled data where the later one is present in large proportion. This Semi-Supervised data can be expressed as

$$D = (D^L, D^U) \tag{1}$$

The focus of Semi-Supervised Learning is to make use of D^L and D^U and come up with a model which will perform better than that with only D^L or only D^U

5.1 Binary Data

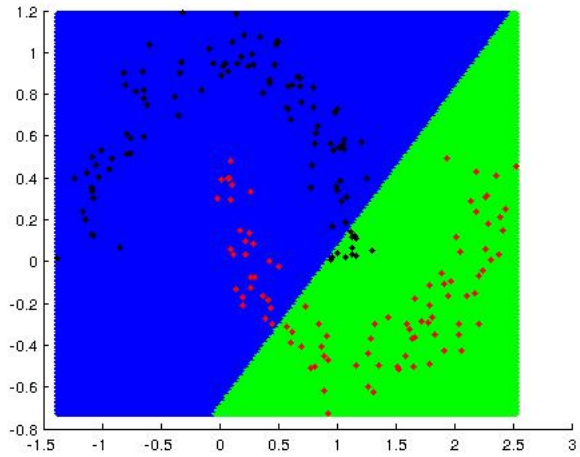
The binary data provided is a 'Two Moons data' which has 200 data points. The data looks like :



5.1.1 Self Training using nu SVM

5.1.1.1 Experiments

1. Only one labeled data point of each class was taken as labeled train data. 70% of the remaining data was taken as a Unlabeled train data and 30% as a test data.
2. The model was chosen by tunning the parameters.
3. The best model we got for the following parameters :
   γ (g) : 0.00095
   cost (c) : 700
   ν (n) : 0.3
   Accuracy = 52.74%
   Linear Boundary

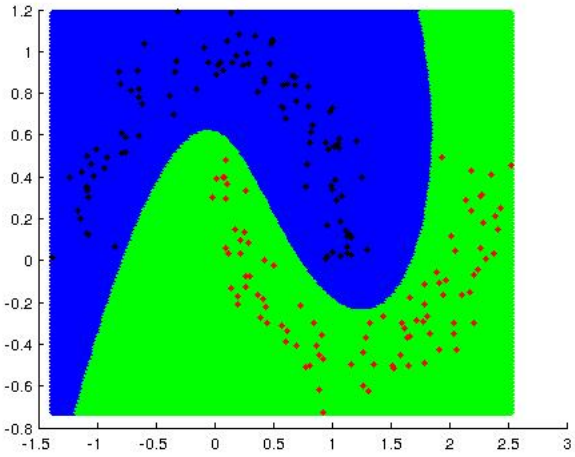


4.

5.1.2 Graph Based Semisupervised Learning using label propagation

5.1.2.1 Experiments

1. Only one labeled data point of each class was taken as labeled train data. 70% of the remaining data was taken as a Unlabeled train data and 30% as a test data.
2. We used ϵ neighborhood method to get the similarity measure.
3. The model was chosen by choosing a proper ϵ value.
4. The best model we got for the following parameters :
   ϵ (g) : 0.42
   Accuracy = 100%



Non Linear Boundary

5.

5.1.3 Semi Supervised SVM

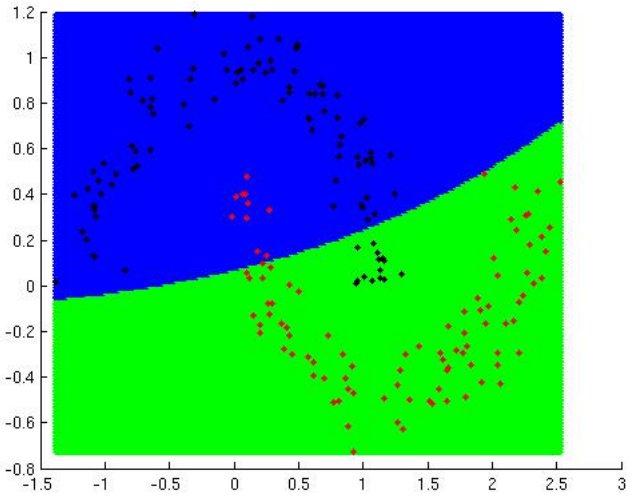
5.1.3.1 Experiments

1. Only one labeled data point of each class was taken as labeled train data. 70% of the remaining data was taken as a Unlabeled train data and 30% as a test data.

2. The model was chosen by tunning the parameters.

3. The best model we got for the following parameters :
   γ (g) : 0.01
   cost (c) : 15

   Accuracy = 81.36%



Non Linear Boundary

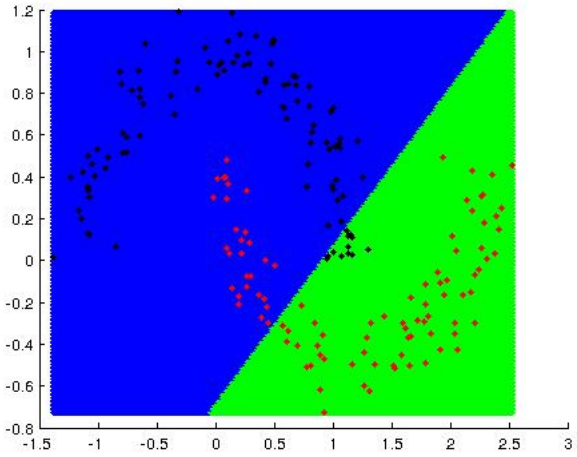
5.1.4 Supervised SVM

5.1.4.1 Experiments

1. Only one labeled data point of each class was taken as labeled train data. 70% of the remaining data was taken as a Unlabeled train data and 30% as a test data.

2. The model was chosen by tunning the parameters.

3. The best model we got for the following parameters :
   γ (g) : 0.01
   cost (c) : 15
   Accuracy = 52.%



Linear Boundary

5.2 UCI Data

The 16 dimensional data provided is a 'letter data' which has nearly 1500 data points. Out of that e used 10% as labelled,60% unlabelled train and 30% test.

For Train Size =10%

Classification Method	Accuracy
Self Training	98.3412
Graph Based Label Propagation	90.0474
SVM Light	97.39
Supervised nu SVM	97.2891
For Train Size =5%	

For Train Size =5%

Classification Method	Accuracy
Self Training	94.1573
Graph Based Label Propagation	90.0474
SVM Light	81.39
Supervised nu SVM	95.2809

For Train Size =30%

Classification Method	Accuracy
Self Training	98.7805
Graph Based Label Propagation	99.0854
SVM Light	97.39
Supervised nu SVM	98.7805

6 Inferences

1. Label Propagation performs the best among all the methods in Binary Data.
2. Selftraining should perform better than Supervised SVM but in these experiments, Supervised performed better.
3. In case of Binary data, performance of supervised and semisupervised was the same. This is because we had only 2 data points and for them we get a linear boundary but as we go for Self training the confidence criteria is not passed by any of the points as both the class conditional probabilities for data are 0.5.
4. SVM light gave fairly good results than self training.

6 Problem-5 : Structured data classification

6.1 Task

The chosen task is to perform "classification" on Graphs for activity against non-small cell lung cancer and ovarian cancer cell lines.

6.2 Dataset

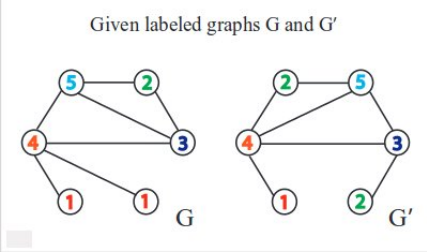
- The dataset NCI1 represent two balanced subsets of data sets of chemical compounds screened for activity against non-small cell lung cancer and ovarian cancer cell lines respectively (Wale and Karypis (2006) and `http://pubchem.ncbi.nlm.nih.gov`).
- The dataset contain 4110 instances of chemical compound activities represented as Graphs.
- The dataset is splitted into 70% as Training data, 10% as Validation data, 20% as Test data.

6.3 Kernel

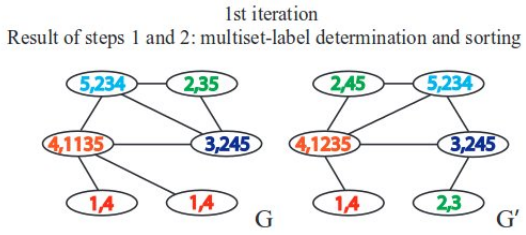
We use one of the graph kernel from Weisfeiler-Lehman (WL) Kernels family to define the similarity between two graphs. The kernel is called as "WL shortest path kernel". In essence, the shortest path kernel counts pairs of shortest paths with the same distance between identically labeled source and sink nodes on the original graphs.

All of the Weisfeiler-Lehman (WL) Kernels for given two graphs G, G' consists of basic 4 steps. as below:

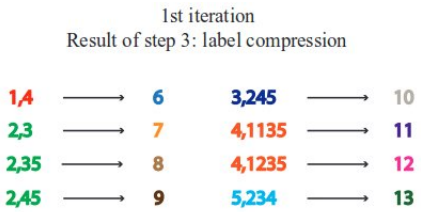
Let the graphs be G, G', as shown in the picture.



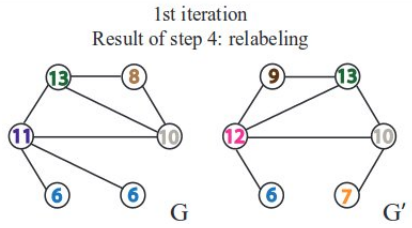
- **Multiset-label determination** : assign a multiset-label to each node in G and G' which consists of the multiset of neighborhood nodes.
- **Sorting each multiset** : Sort elements in the Multiset-label and concatenate them into a string.



- **Label compression** : Sort all of the strings for all $v \in G \& G'$ in ascending order. Map each distinct string into a new label using function $f : \sum^* \rightarrow \sum$, where $f(string(v)) = f(string(w))$, iff, $string(v) = string(w)$



- **Relabeling** : for all nodes in graphs G, G', set the label as $f(string(v))$



These four steps are repeated to get a final relabeled graph from which the Kernel's feature maps are derived.

6.3.0.1 Base kernel

we consider the base kernel k_{SP} of the form k_{SP}(G, G') = $\phi_{SP}(G)^T \phi_{SP}(G')$, where $\phi_{SP}(G)^T$ is a vector whose components are number of occurences of triplets of the form $< a, b, p >$ in G, where a,b $\in \sum$ are ordered end-point labels of shortest path and p $\in N_0$ is the shortest path length. The base kernel at each iteration i, can be combined together to give out the final kernel.

$k^h_{WLshortestpath} = k_{SP}(G_0, G'_0) + k_{SP}(G_1, G'_1) + \cdots + k_{SP}(G_h, G'_h)$

6.4 Confusion matrices and performance details

For the current dataset (NCI1), the iteration count is set as 3. The base kernel matrices for all the three iterations are retrieved and added to get the actual kernel $k^h_{WLshortestpath}$.

The best performance is achieved with $\nu = 0.3$. In the best model, the number of bounded SVs = 422, number of unbounded SVs = 1255.

6.4.1 Validation data

	non-small cell lung cancer (predicted)	ovarian cancer cell (predicted)
non-small cell lung cancer (Target)	181	24
ovarian cancer cell (Target)	30	175

	Formula	Value
True positive rate w.r.t., Normal class (recall)	$\frac{TP}{TP+FN}$	0.854
False positive rate w.r.t., Normal class (fall-out)	$\frac{FP}{TN+FP}$	0.218
F1-score	$\frac{2TP}{2TP+FN+FP}$	0.866
Accuracy		86.82%

6.4.2 Test data

	non-small cell lung cancer (predicted)	ovarian cancer cell (predicted)

non-small cell lung cancer (Target)	338	75
ovarian cancer cell (Target)	54	357

	Formula	Value
True positive rate w.r.t., Normal class (recall)	$\frac{TP}{TP+FN}$	0.868
False positive rate w.r.t., Normal class (fall-out)	$\frac{FP}{TN+FP}$	0.291
F1-score	$\frac{2TP}{2TP+FN+FP}$	0.846
Accuracy		84.34%