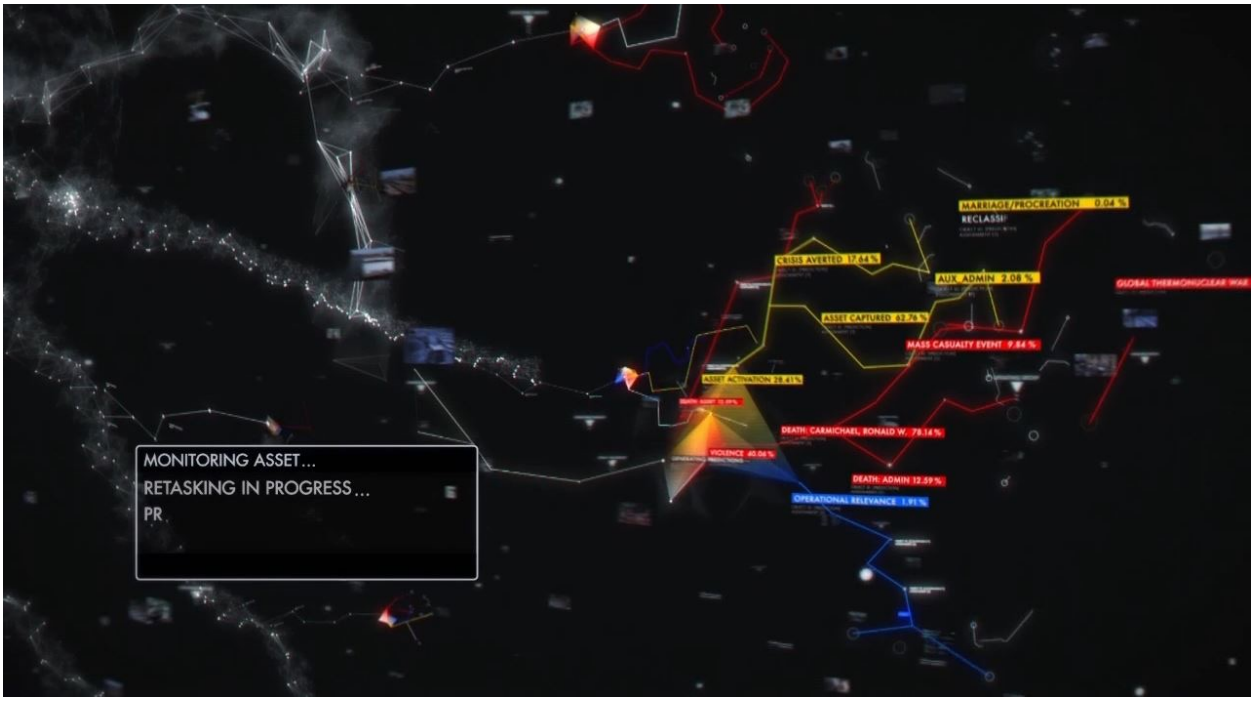
AIML Report

Support Vector Machines



Ramnath Pillai

Fall 2015

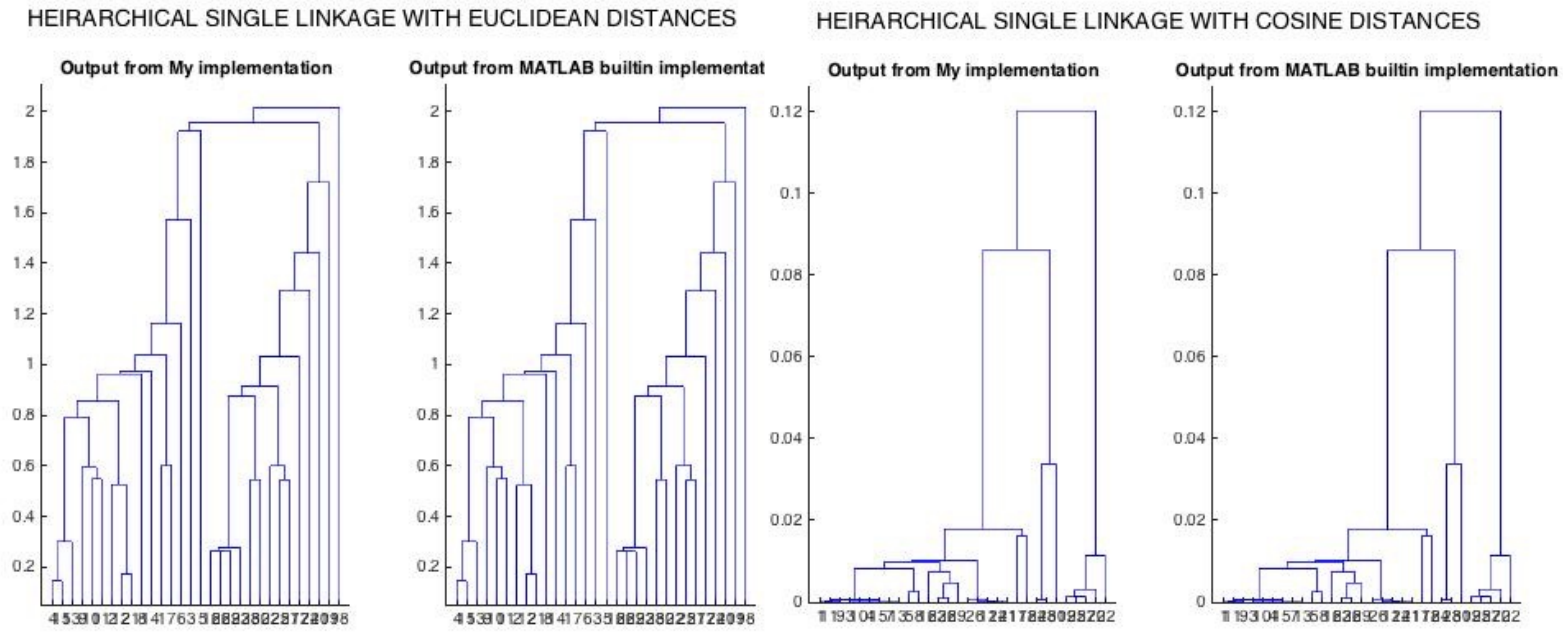
Homework 5

Q1. a. CODE TO RUN: main_singlelink.m, main_completelink.m and main_averagelink.m

CODE FUNCTIONS HAVING IMPLEMENTATIONS: cluster_singleLink.m, cluster_completeLink.m, cluster_averageLink.m

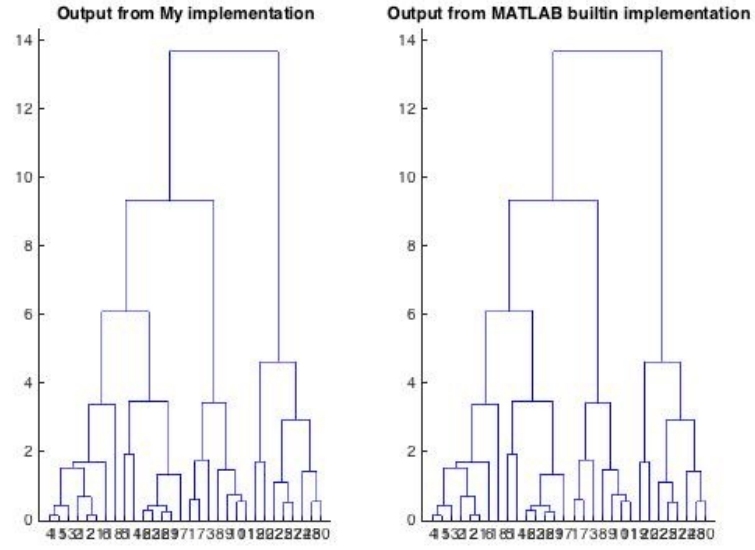
(It was extremely difficult to integrate this into a single file)

FOR SINGLE LINK:

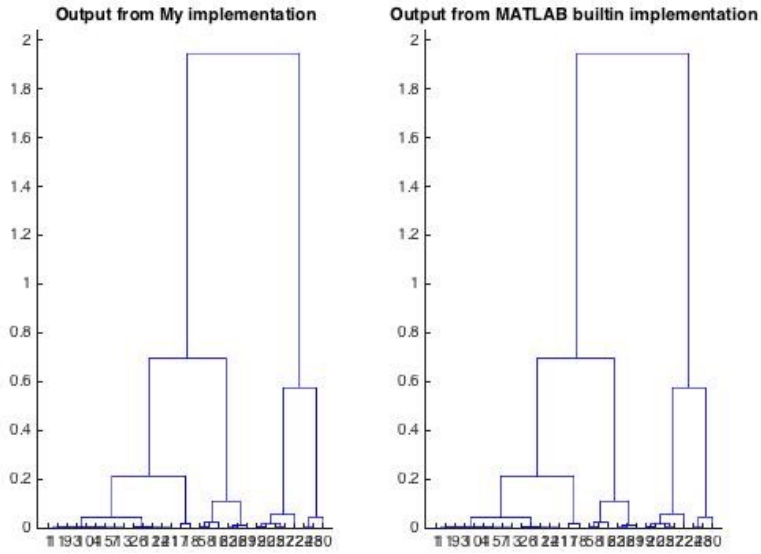


FOR COMPLETE LINKAGE:

HEIRARCHICAL COMPLETE LINKAGE WITH EUCLIDEAN DISTANCES

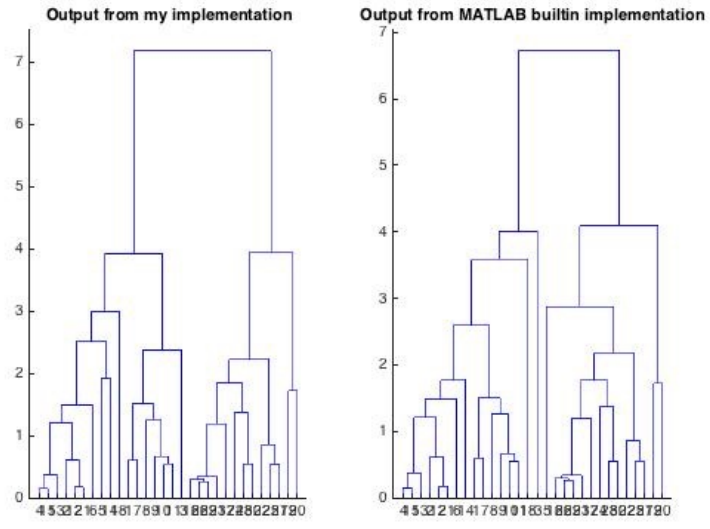


HEIRARCHICAL COMPLETE LINKAGE WITH COSINE DISTANCES

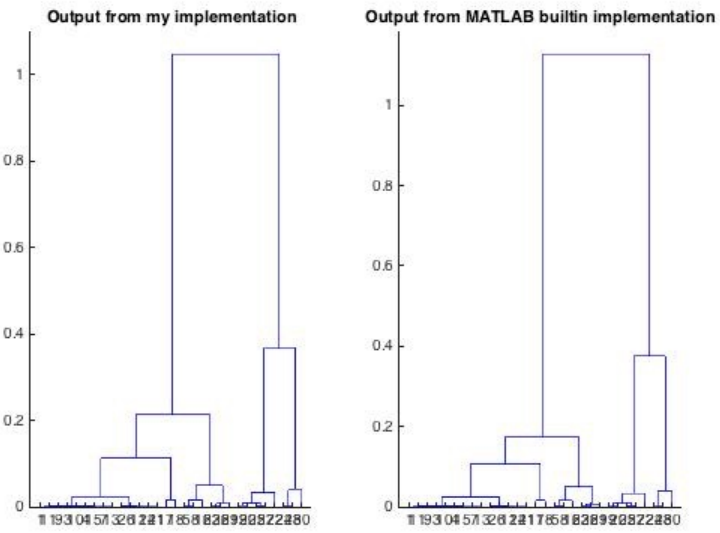


FOR AVERAGE LINKAGE:

HEIRARCHICAL AVERAGE LINKAGE WITH EUCLIDEAN DISTANCES



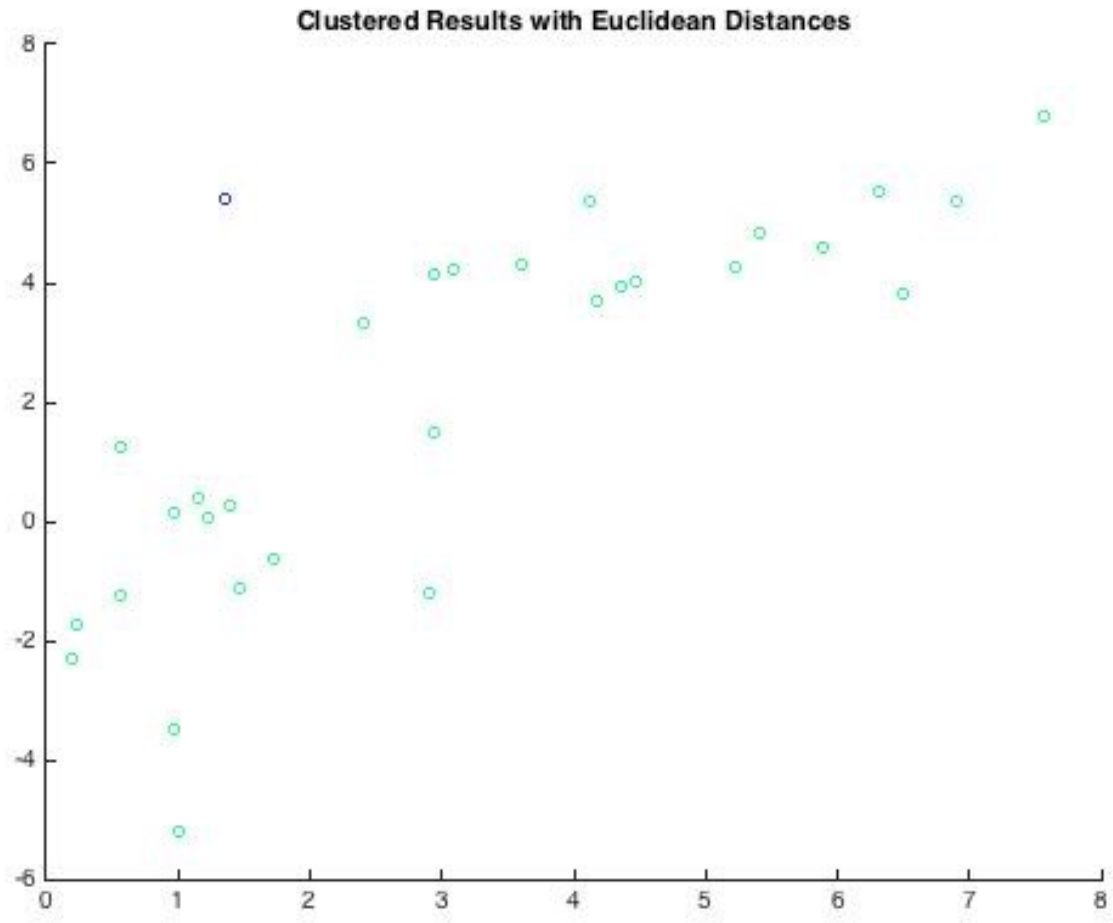
HEIRARCHICAL AVERAGE LINKAGE WITH COSINE DISTANCES



Q1.b

As it is not feasible to organize the data into a small table due to constraints of space, I have displayed the results in an intuitive sequence.

FOR SINGLE LINKAGE, EUCLIDEAN DISTANCES



Indices in one cluster are

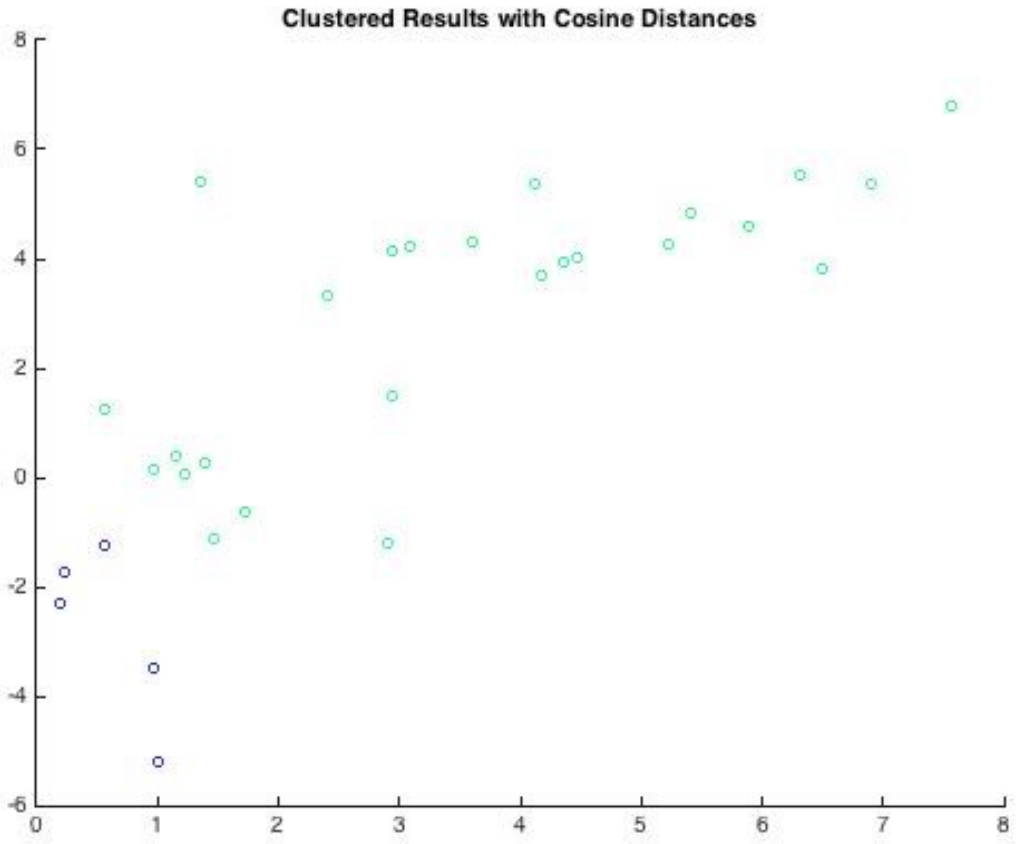
I = 18

Indices in other cluster are

I = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 19 20 21 22 23 24 25 26 27 28 29 30

Accuracy with the ground truth for this case is **53.333333 percent**

SINGLE LINKAGE COSINE DISTANCES:



Indices in one cluster are

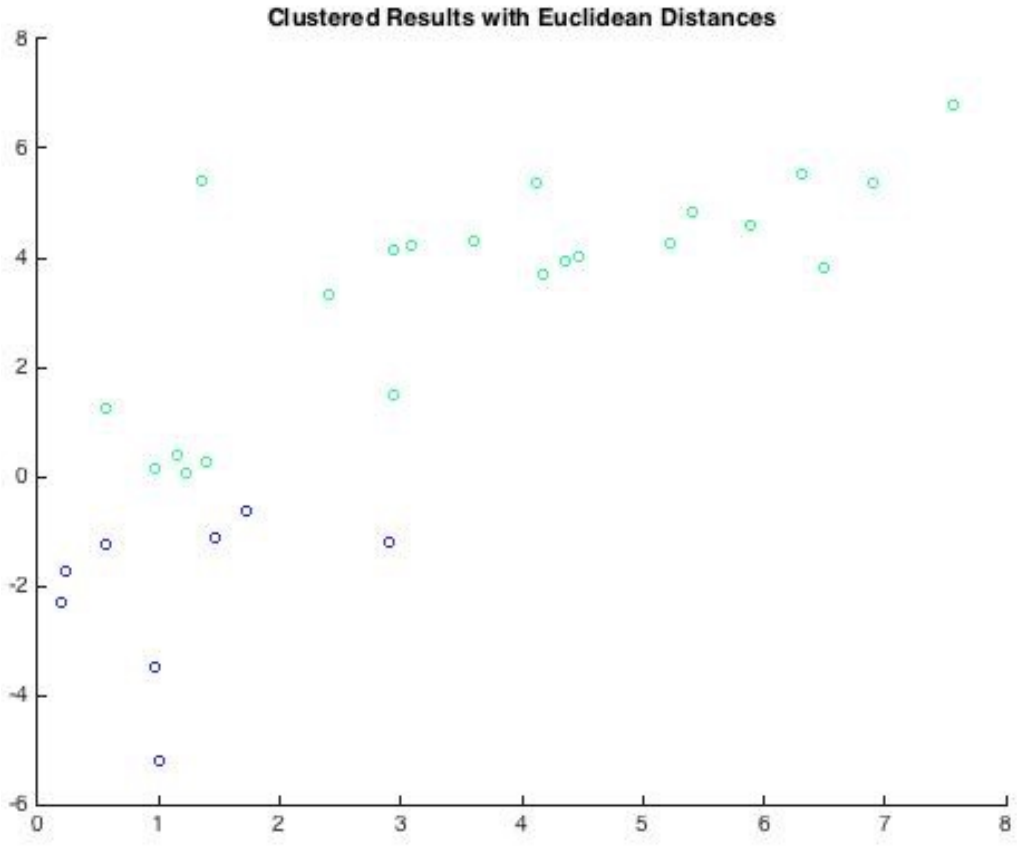
I =19 20 22 25 27

Indices in other cluster are

I = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 21 23 24 26 28 29 30

Accuracy with the ground truth for this case is **66.666667 percent**

COMPLETE LINKAGE WITH EUCLIDEAN DISTANCE:



Accuracy with the ground truth for this case is 76.666667 percent

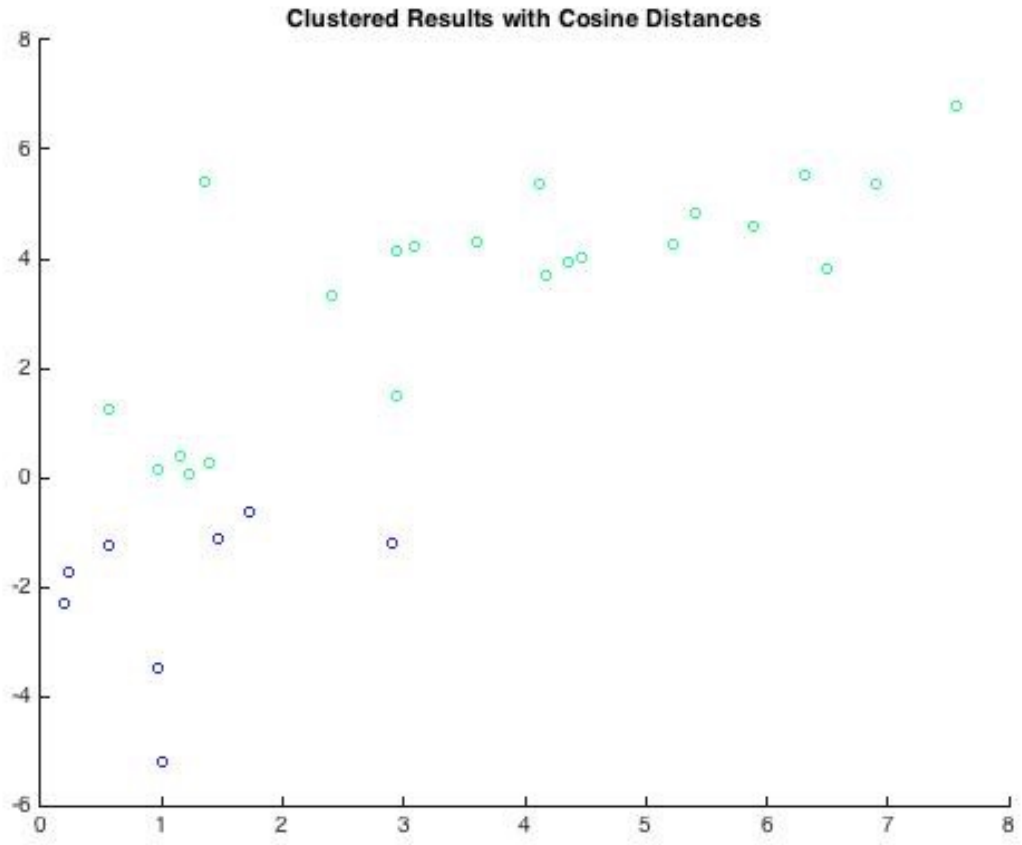
Indices in one cluster are

I =19 20 22 24 25 27 28 30

Indices in other cluster are

I =1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 21 23 26 29

COMPLETE LINKAGE WITH COSINE DISTANCE:



Accuracy with the ground truth for this case is 76.666667 percent

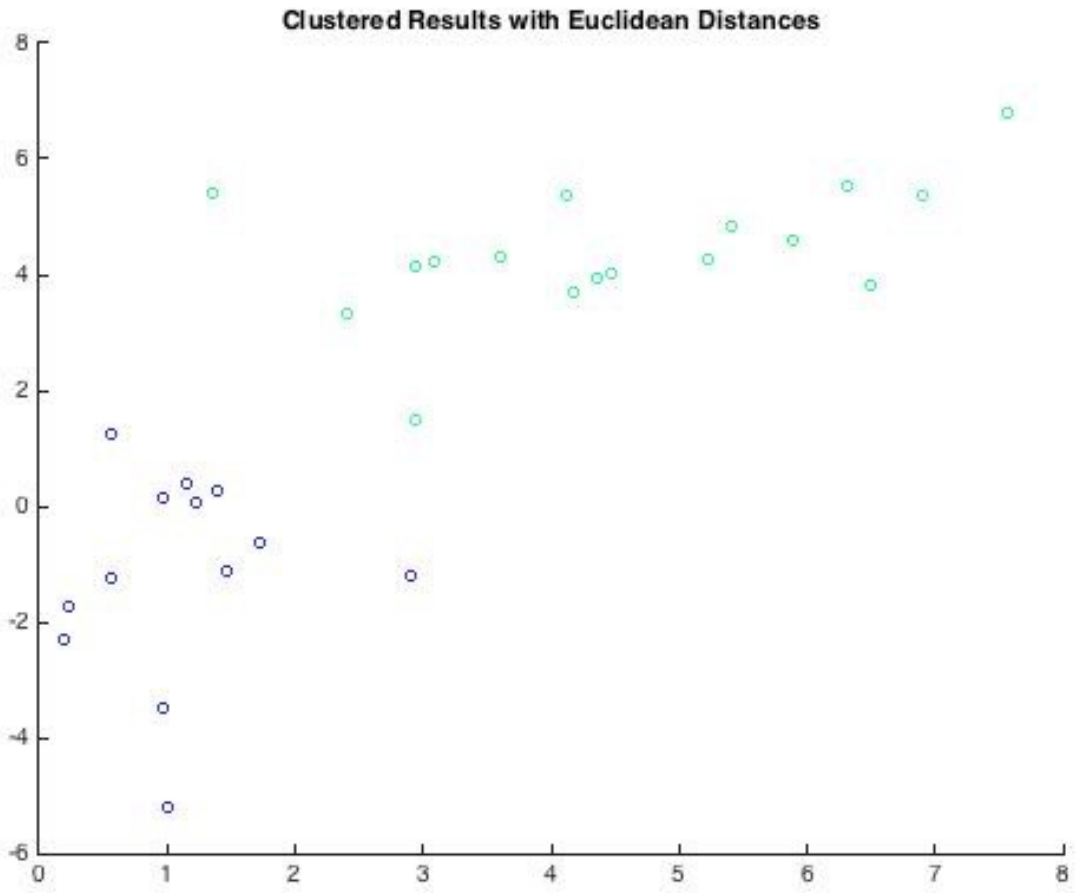
Indices in one cluster are

I =19 20 22 24 25 27 28 30

Indices in other cluster are

I =1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 21 23 26 29

AVERAGE LINKAGE WITH EUCLIDEAN DISTANCE:



Accuracy with the ground truth for this case is 93.333333 percent

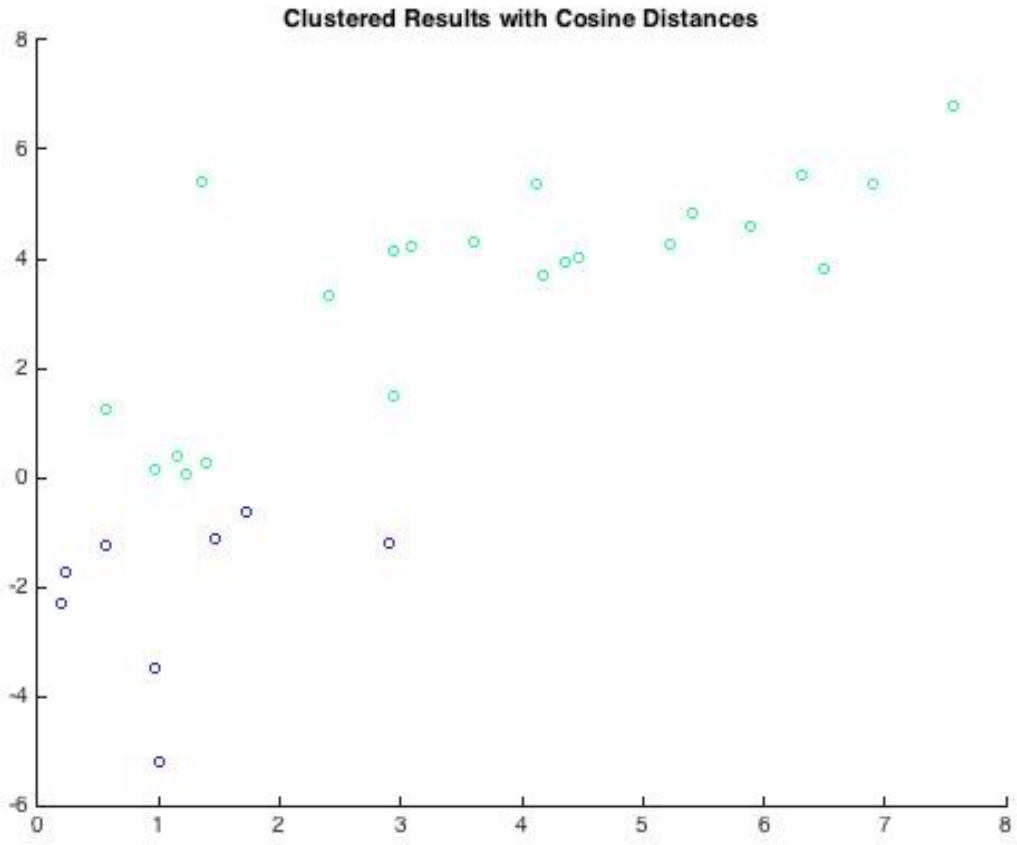
Indices in one cluster are

I =16 17 19 20 22 23 24 25 26 27 28 29 30

Indices in other cluster are

I =1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 18 21

AVERAGE LINKAGE WITH COSINE DISTANCE:



Accuracy with the ground truth for this case is 76.666667 percent

Indices in one cluster are

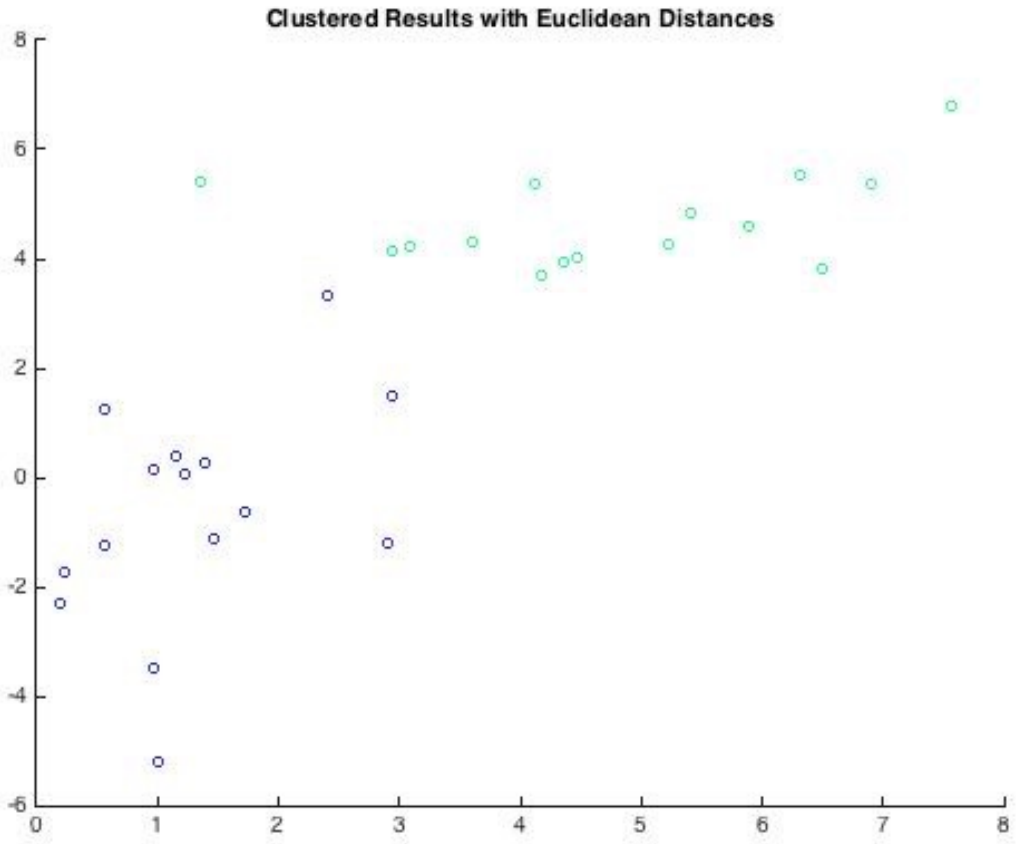
I =19 20 22 24 25 27 28 30

Indices in other cluster are

I =1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 21 23 26 29

Q1.c CODE: mykmeans.m

K MEANS WITH EUCLIDEAN DISTANCES



Accuracy with the ground truth for this case is 86.666667 percent

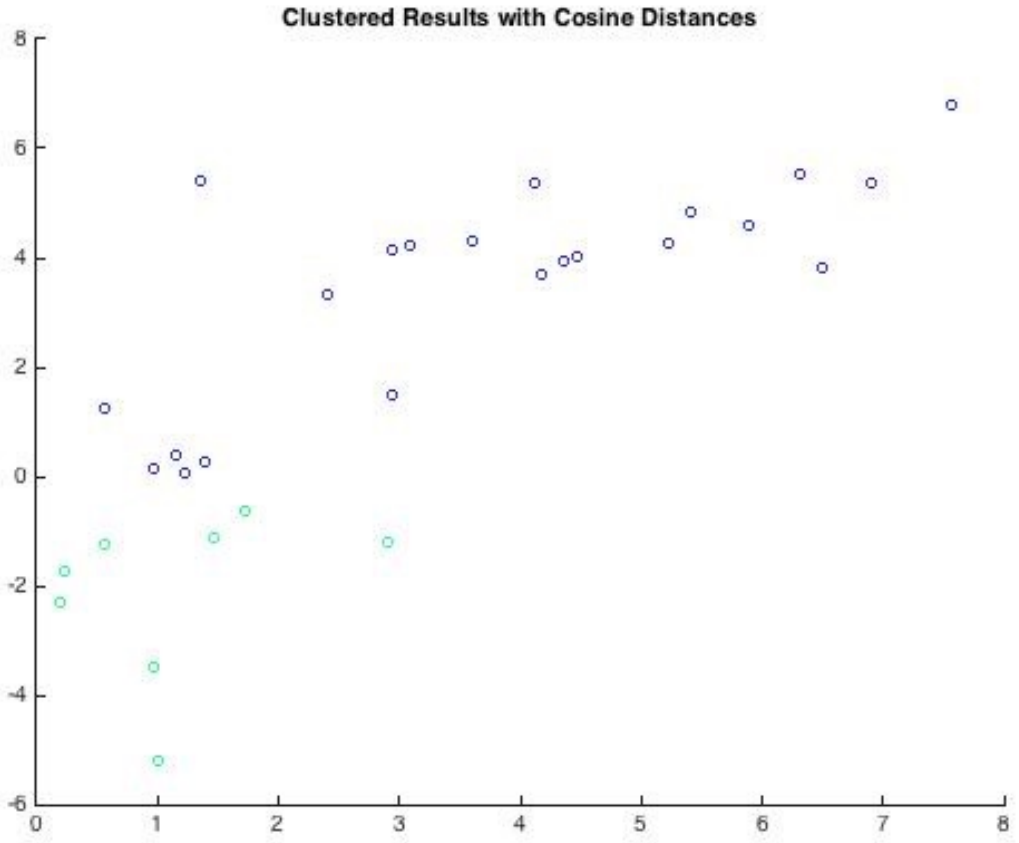
Indices in one cluster are

I = 5 14 16 17 19 20 22 23 24 25 26 27 28 29 30

Indices in other cluster are

I =1 2 3 4 6 7 8 9 10 11 12 13 15 18 21

K MEANS WITH COSINE DISTANCES



Accuracy with the ground truth for this case is 76.666667 percent

Indices in one cluster are

I = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 21 23 26 29

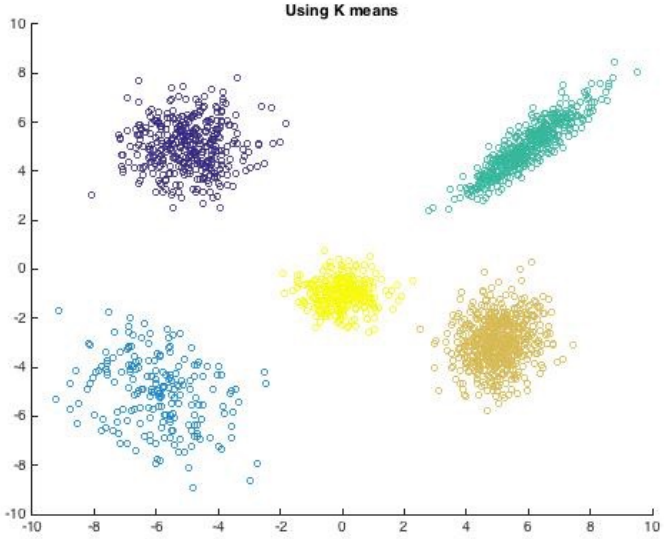
Indices in other cluster are

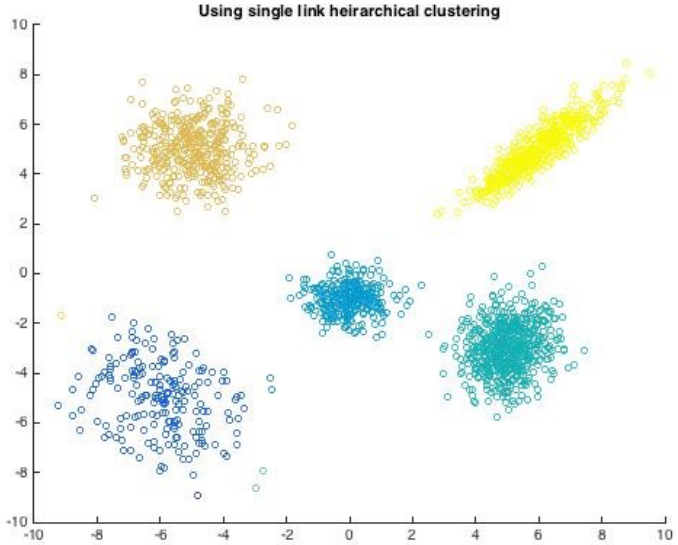
I =19 20 22 24 25 27 28 30

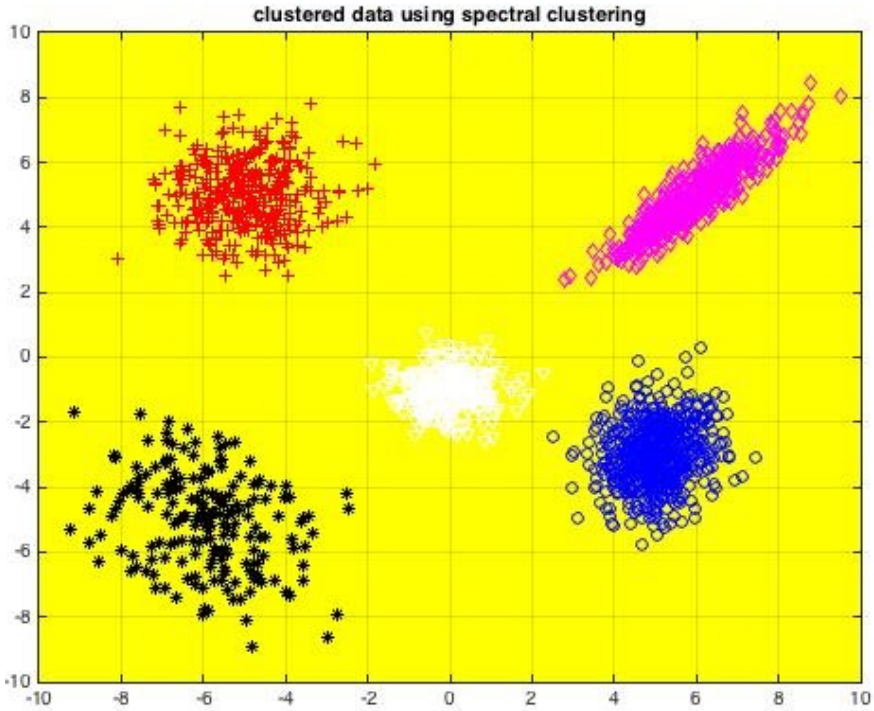
Q2. Code: hw5_clustering.m

NOTE: IN THE SAVED TEXT FILES, FIRST COL: K MEANS, 2ND COL: HIERARCHICAL, 3RD COL: SPECTRAL. ALSO, THE NUMBERING OF THE CLUSTERS IS MEANINGLESS. The aggregation of these numbers gives the cluster info.

DATASET 1:







Consider a quantitative estimate of accuracy metric as "**Total number of completely accurate clusters"/"Total number of clusters"**

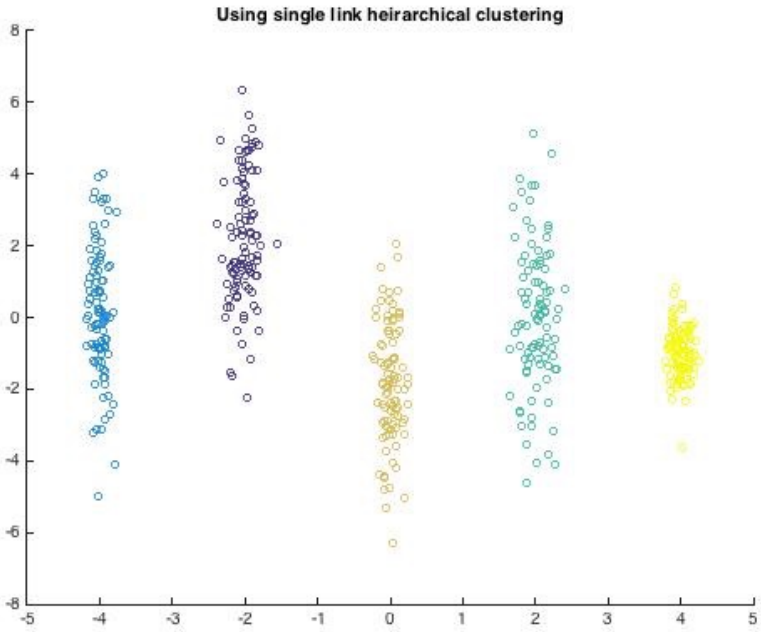
The dataset 1 is split into five distinct intuitive data segregations. It can be seen that K means split the dataset 1 well. With a priori knowledge of the number of clusters, K means shows very good clustering of the data. **Accuracy=100%**

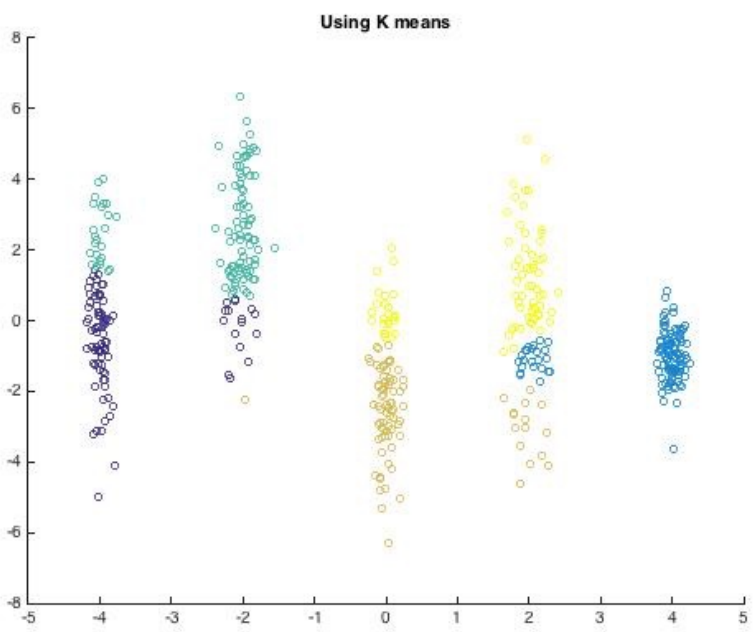
Single link hierarchical clustering also splits the data very well (note that two clusters have similar colors, but they are actually different clusters). Since the single link hierarchical clustering is based on smallest euclidean distance between the various clusters, where every point is a separate cluster initially, it gives very good results as the current datasets have distinct separation of data. **Accuracy=100%**

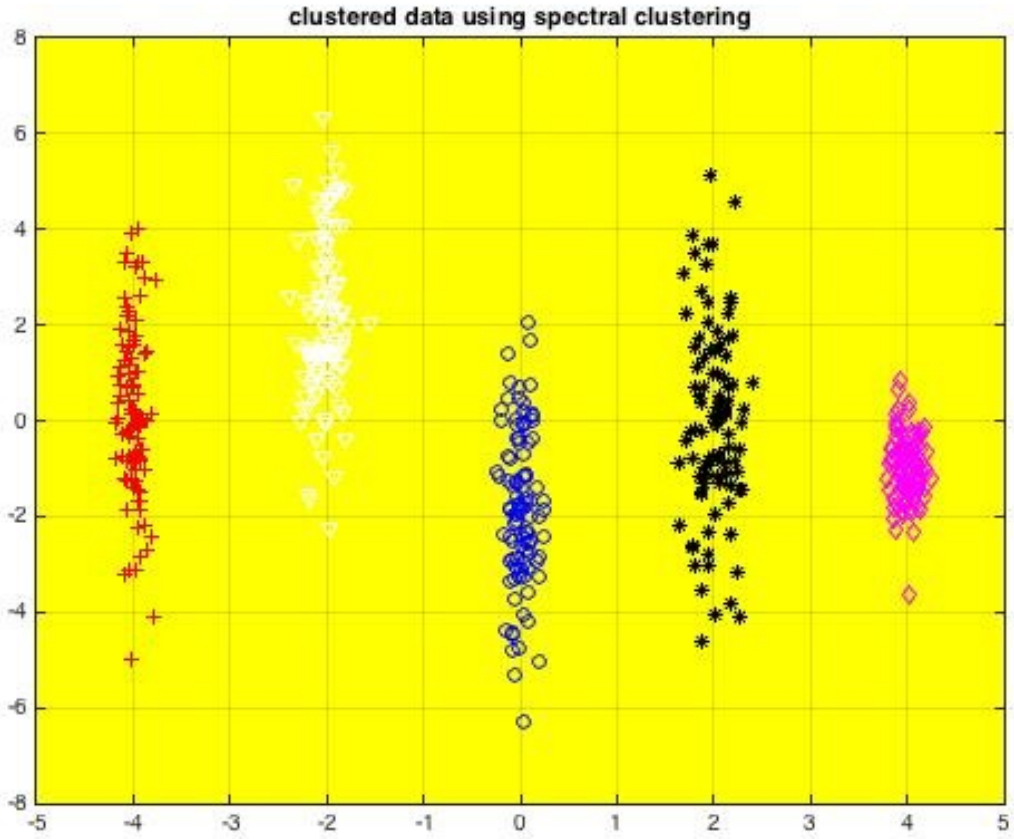
Spectral clustering uses a graph model in order to predict the clusters. As this method uses the concept of association metric between the adjacent points in a cluster, and then employs normalized cut to distinguish between the main clusters, it works extremely well in these cases. **Accuracy=100%**

In this dataset, all three methods perform equally well and cannot make a decision as to which one performs better.

DATASET 2:


Using single link heirarchical clustering


Using K means


clustered data using spectral clustering

Dataset 2 is split into linear segments.

K means: The method of K means fails in this case as the method of K means groups part of the first cluster with a part of the second cluster. When one of the means of the dataset converge lie between two of the clusters, part of both clusters are grouped into a single cluster. Therefore, this method gives a bad clustering. Also, the fact that K means depends on the initial guess for the means, k means is not a good method to classify this data set. **Accuracy=20%**

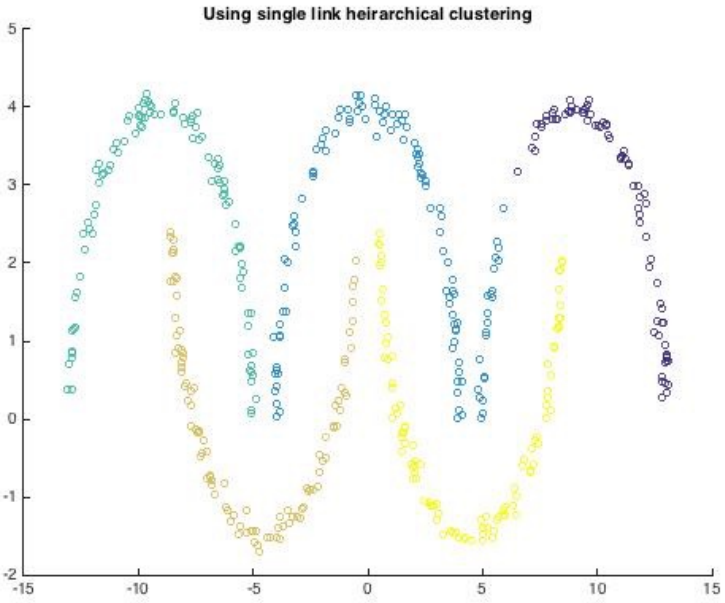
Hierarchical Single link:

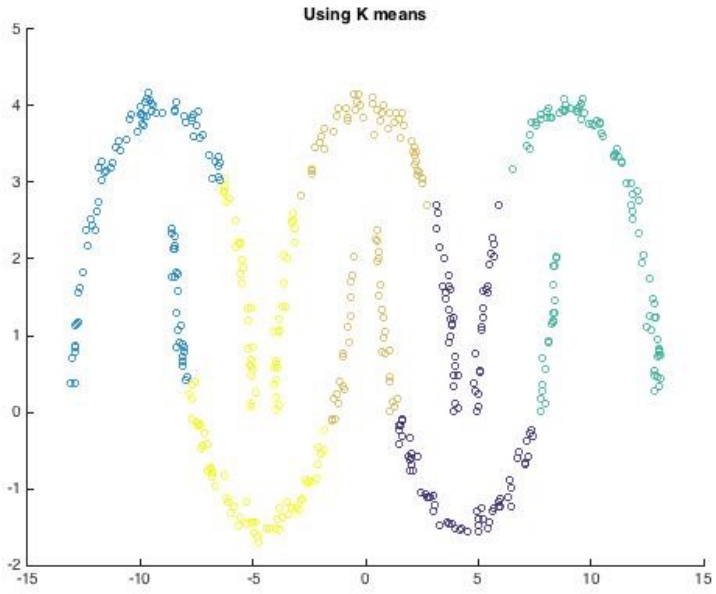
Single link hierarchical clustering splits the data very well. Since the single link hierarchical clustering is based on smallest euclidean distance between the various clusters, where every point is a separate cluster initially, it gives very good results as the current datasets have distinct separation of data. **Accuracy=100%**

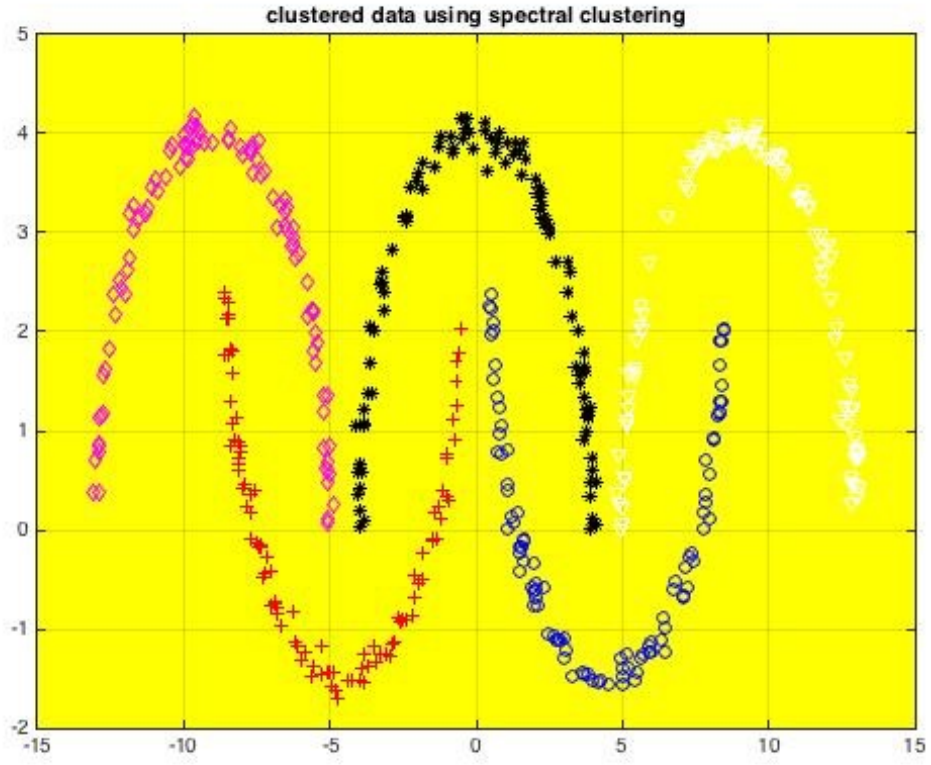
Spectral Clustering:

Spectral Clustering uses a graph model in order to predict the clusters. As this method uses the concept of association metric between the adjacent points in a cluster, and then employs normalized cut to distinguish between the main clusters, it works extremely well in these cases. **Accuracy = 100%**

DATASET 3:


Using single link heirarchical clustering


Using K means


clustered data using spectral clustering

K Means: It can be seen that as the current dataset is based on the mutual association between adjacent data points based on a distance metric, K means fails to aggregate the data into proper clusters. K means is based on grouping data over a certain radius of area, therefore it is not a good method to cluster data such as this set. **Accuracy=0%**

Single Link Hierarchical Clustering:

This splits the data fairly well (note that there are 2 very similar colors used in the clusters, which are actually different, but at first sight appears to be same), but fails at places where the points belonging to the same cluster have points that are farther than the usual trend till that point. This is evident from the third crescent on the top row. This would be a good method if the points are uniformly distributed in curves as shown above, but may give bad aggregation in case the points exhibit noise. **Accuracy=80%**

Spectral Clustering: Spectral Clustering uses a graph model in order to predict the clusters. As this method uses the concept of association metric between the adjacent points in a cluster, and then employs normalized cut to distinguish between the main clusters, it works extremely well in these cases. **Accuracy=100%**

OVERALL CONCLUSIONS:

Spectral Clustering is consistently a better method than the others. This is due to the same reasons as explained in all the above cases.

SUPPORT VECTOR MACHINES:

Dataset 1 can be perfectly classified by linear SVMs as they could be segregated by lines.

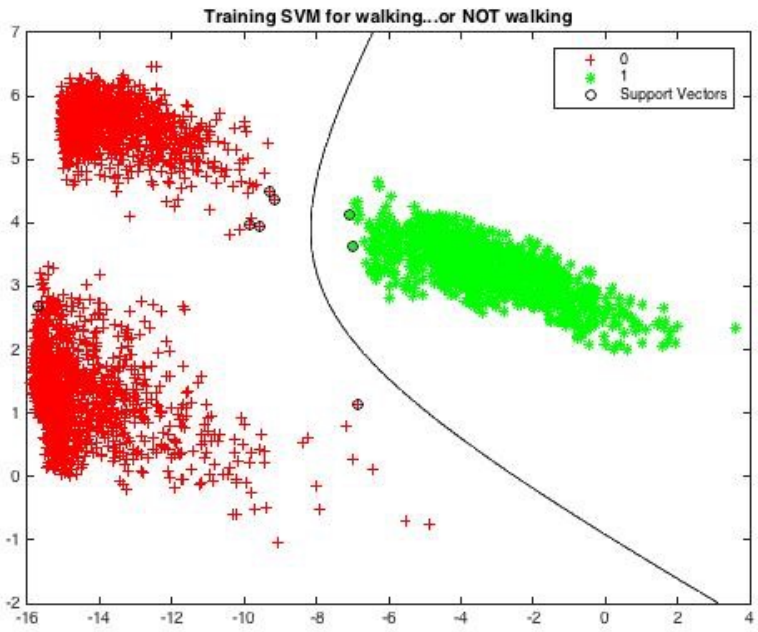
Dataset 2 can be segregated using linear SVMs as this dataset can also be perfectly separated using lines.

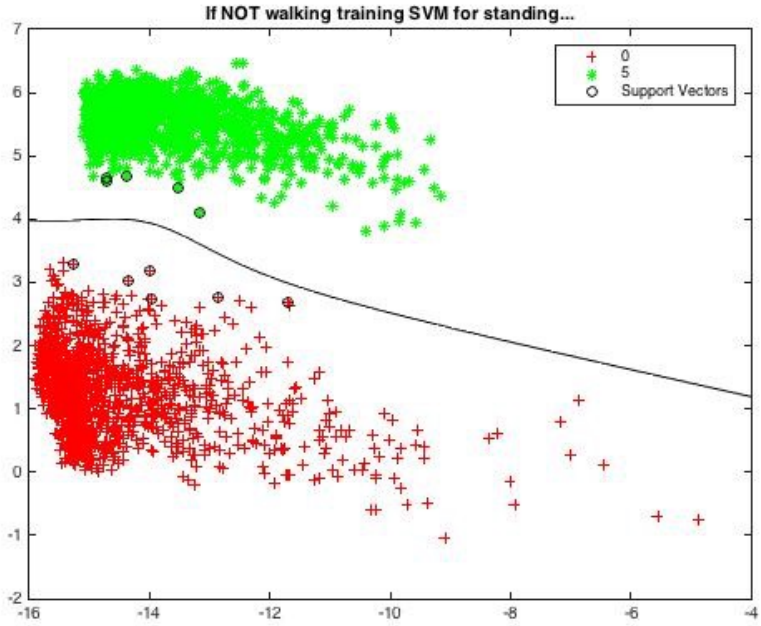
Dataset 3 cannot be segregated using linear SVMs. However a clustering/learning can be obtained using a high dimensional SVM segregation using a custom Kernel that helps to obtain appropriate closed curves that separate these curves.

Q3. CODE: **SVMhw5_SVM_activity_recognition.m for separating 3 classes with 2 PCs.**

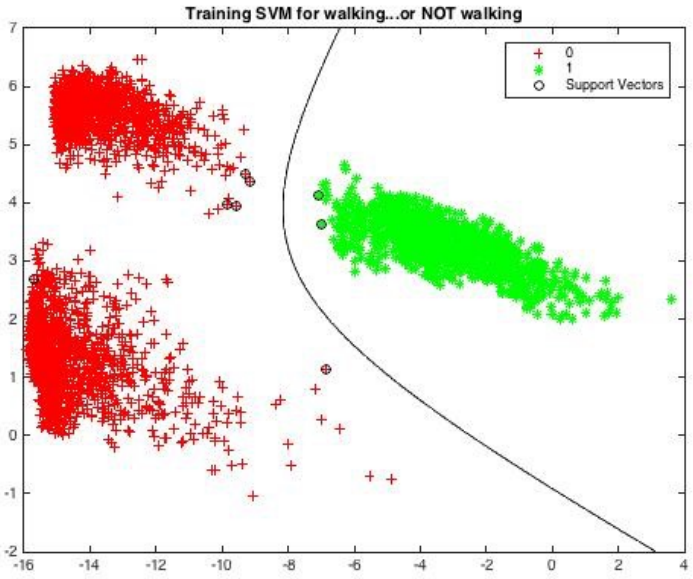
First the data was classified based on 3 SVMs based one versus all strategy. It yielded an accuracy of 95%. But it was noticed that in this particular question, the attribute for "walking" neatly split the data into two. Therefore, this data was classified based on 2 SVMs by filtering out the data selectively and passing on to the next SVM. This one has an accuracy of 99.5%

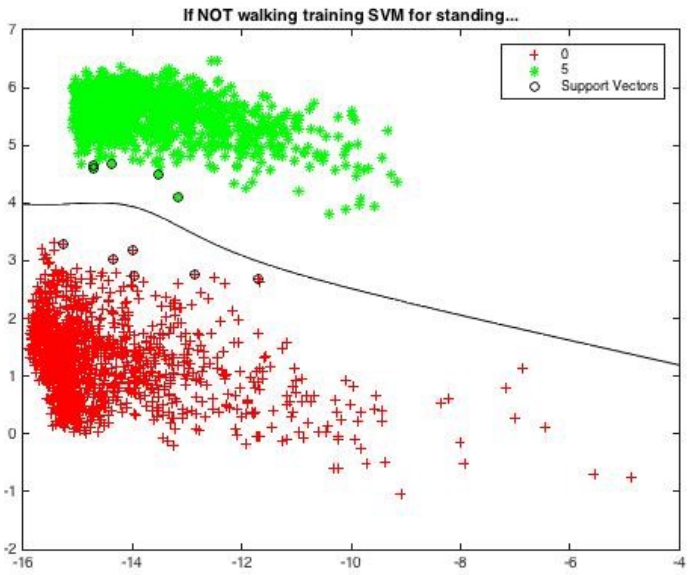
Firstly, the SVM was trained for recognizing walking. The non-walking data was then trained to recognize standing or laying. The 2 SVMs corresponding to this split are shown below:





Since this is a clean split, it is counter productive to train the next SVM with the entire dataset. Thus the next SVM was trained using the NON walking dataset (RED). Both the SVMs use a polynomial kernel of cubic order as this was found to effectively learn the above distribution. The overall SVM behavior including the testing data is shown in the plots below:





- IF AN EXTRA POINT (4,-1) was introduced into the dataset, there will be NO CHANGE in the SVM. As it can be seen the point (4,-1) lies on the right side of the classifier that classifies walking. This is in agreement with the previous decision boundary created. Therefore this point will not act as a support vector and therefore the decision boundary remains same and so does the classification.

- Using the 2 SVM scheme shown above the test accuracy is given to be 99.5%.

confusion =

```
 496    0    0
   0  532    8
   0    0  529
```

Since clustering is a method of UNSUPERVISED LEARNING, the output does not depend on the labels assigned to it. Thus it does not make sense to expect accuracy after using such a training method on a labelled data set. However, on inspection of the dataset, we can notice that there is a visible clustering variation sets. This can be effectively captured using K means. Thus K means would be a good method to classify this data.

The accuracy using K means method is given to be 99.75%. This makes sense as there is a clear spatial clustering of the data when plotted in a 2D coordinate system and the basic of metric of classification could be generalized as a distance metric. Therefore, K means gives an excellent output.

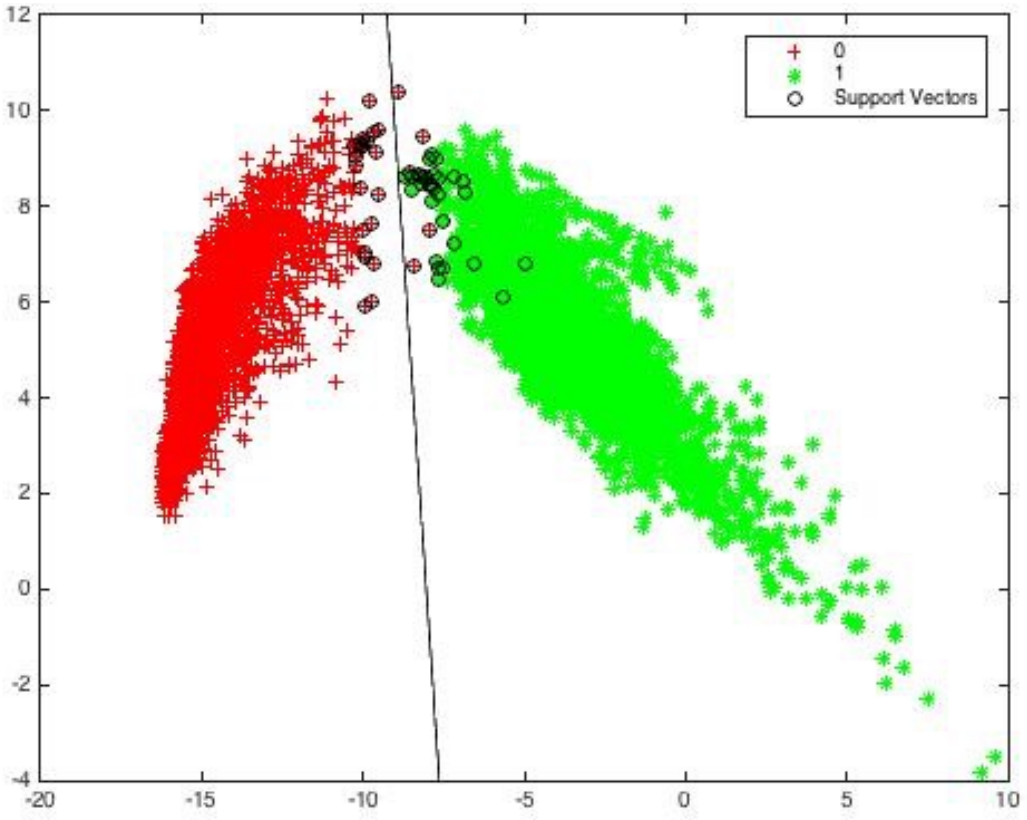
USING ALL THE TRAINING DATA:

CODE: **SVMAll_activity_recognition.m**

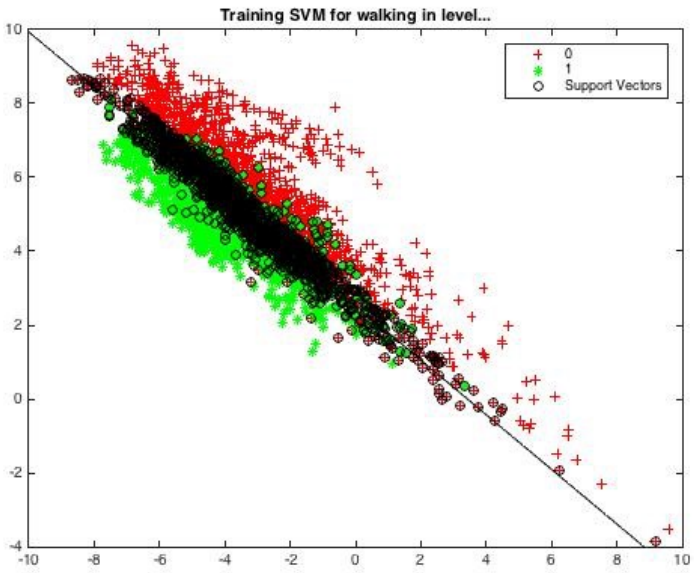
A similar approach to the first case was followed. Here 7 SVMs were used to train this data as it yielded better results by using selective exclusion for the subsequent SVM training data.

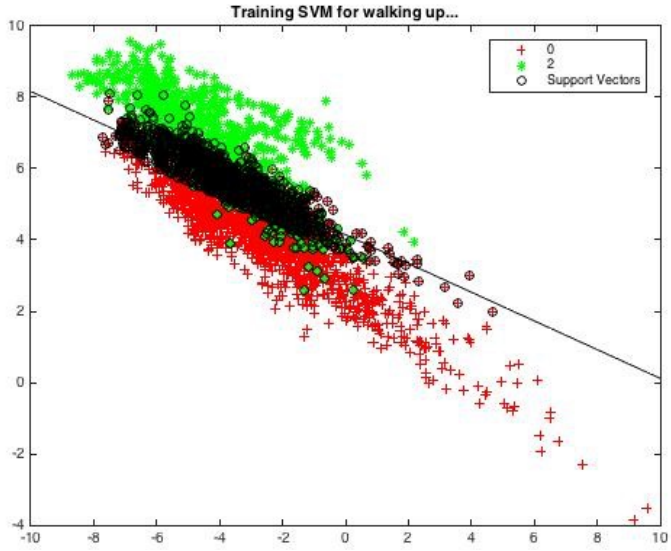
First, an SVM was trained to recognizing all walking models and all non walking models. This model uses a linear Kernel. This provided a clear split of data. All non linear kernels used here are quadratic kernels. After that, each class in it was trained using separate SVMs. Hence a total of 6+1 SVMs. When a 6 SVM approach was followed, there was more confusion in classification compared to this method. The 6 SVM approach yielded an accuracy of 51%.

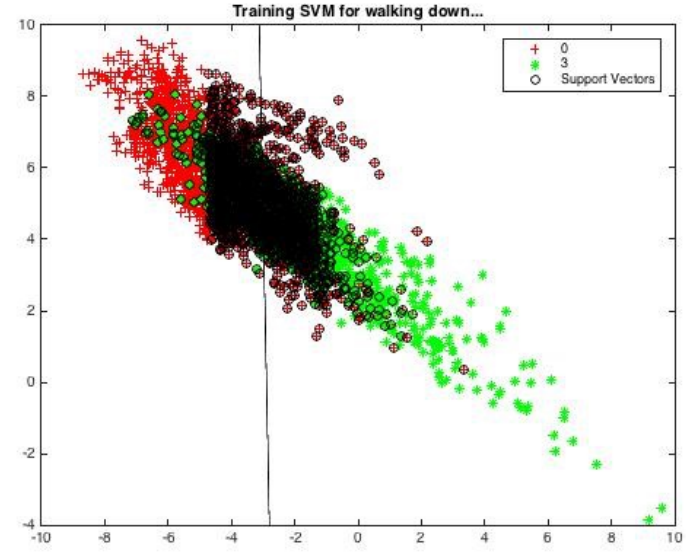
SPLIT BETWEEN WALKING /NON WALKING MODELS

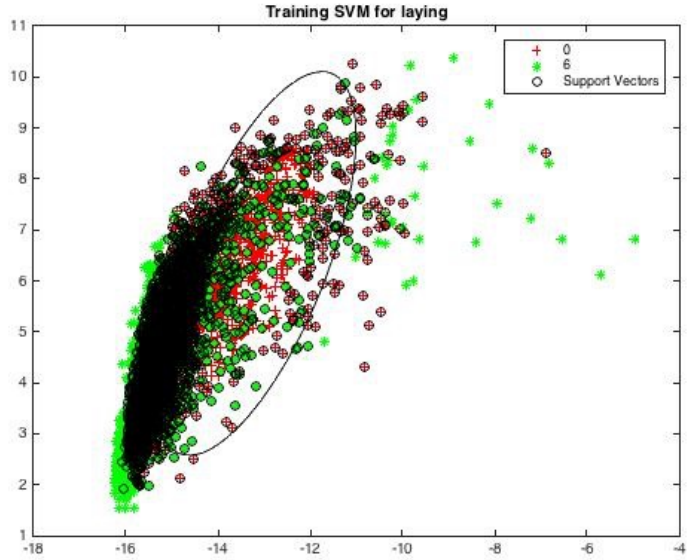


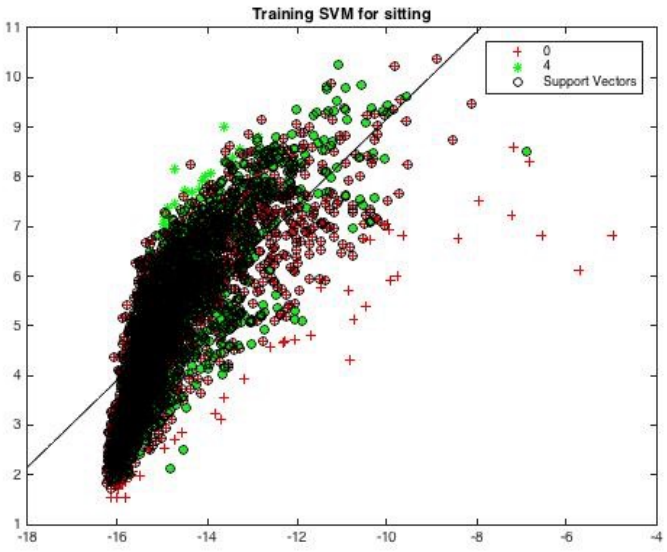
The remaining SVMs are shown below:

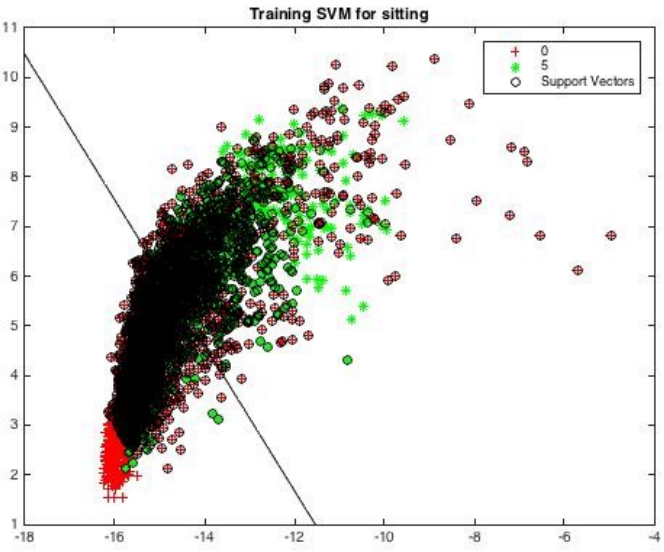




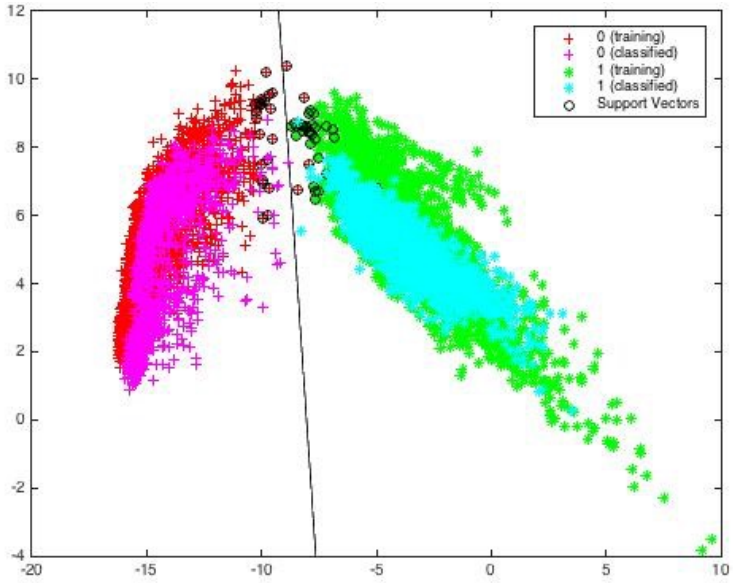


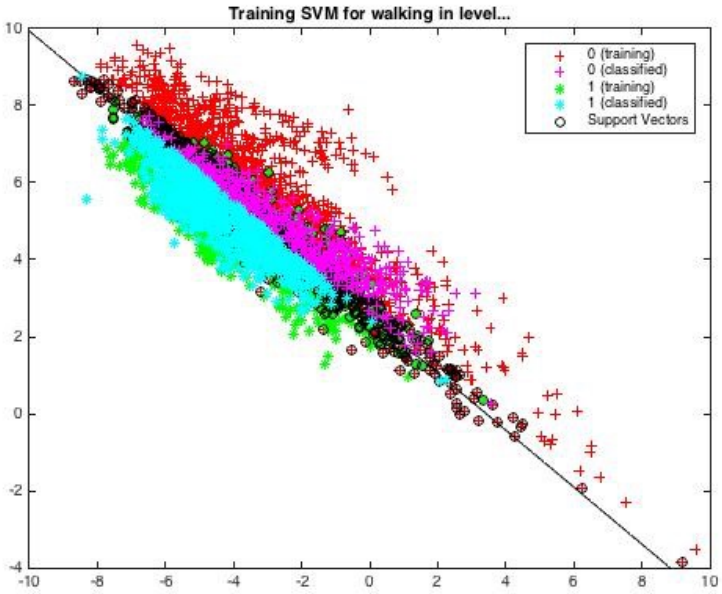


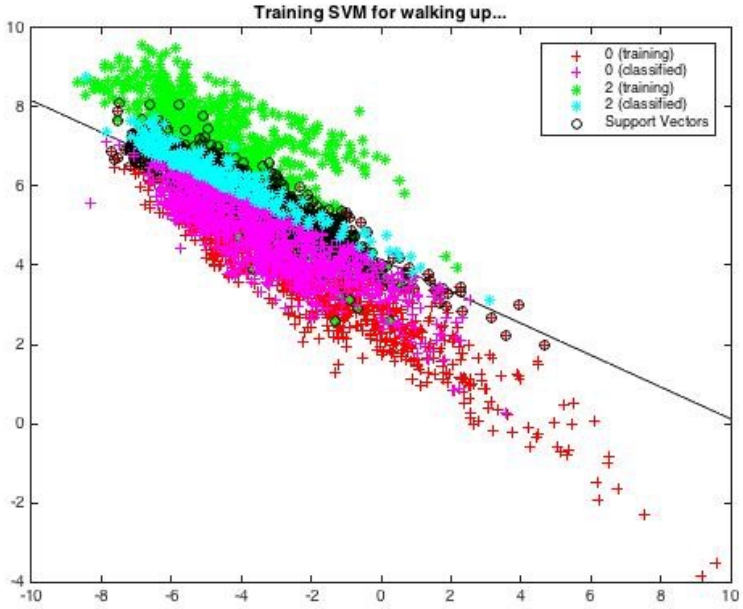


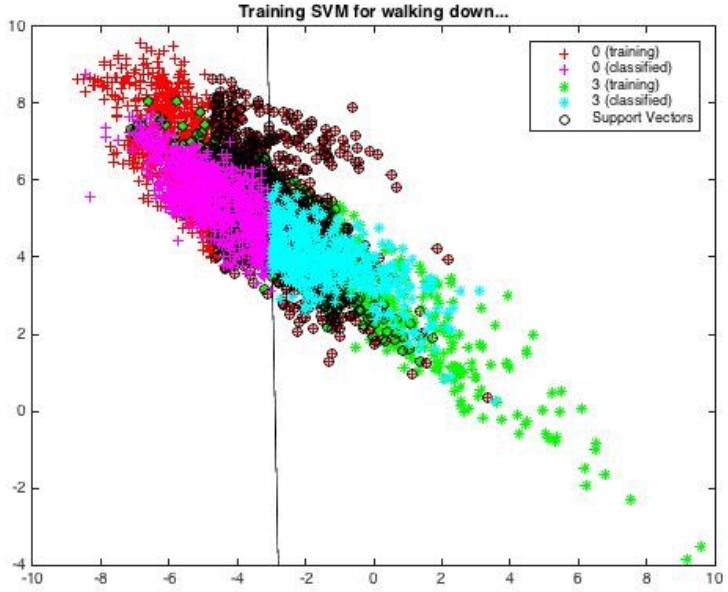


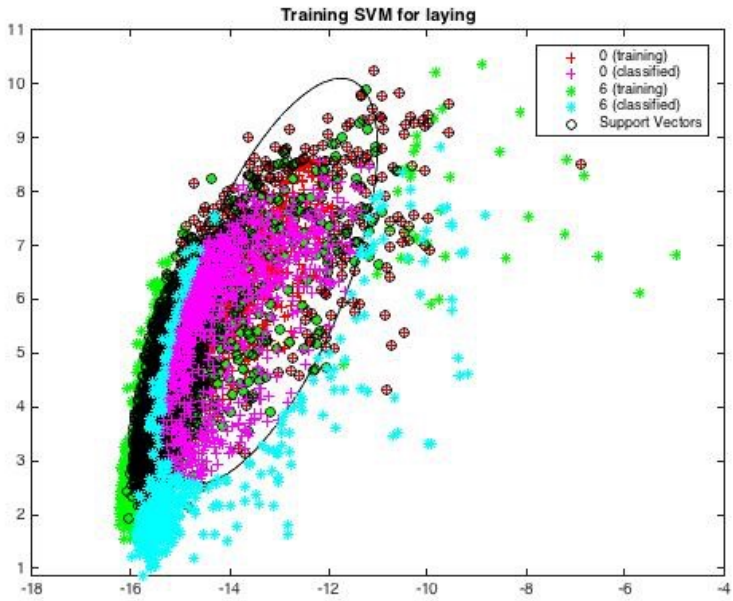
Below is a plot of the classified SVMs

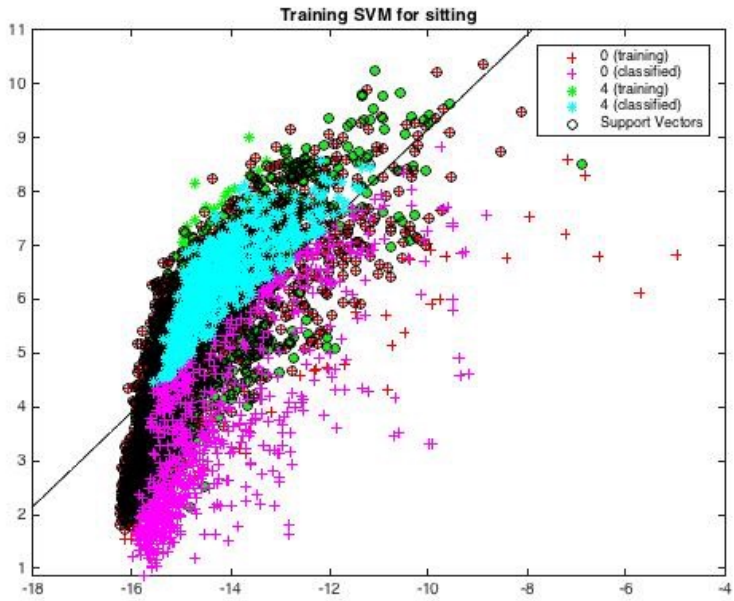

0 (training)
0 (classified)
1 (training)
1 (classified)
Support Vectors

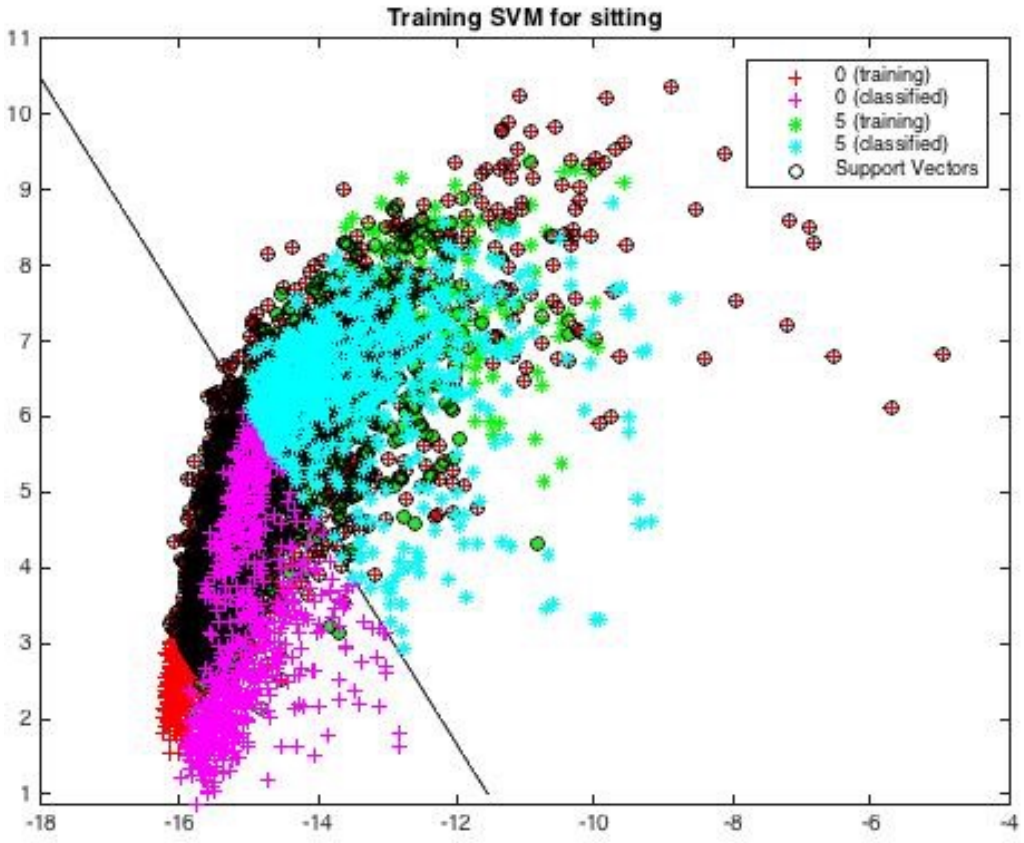

Training SVM for walking in level...
0 (training)
0 (classified)
1 (training)
1 (classified)
Support Vectors


Training SVM for walking up...
0 (training)
0 (classified)
2 (training)
2 (classified)
Support Vectors


Training SVM for walking down...
0 (training)
0 (classified)
3 (training)
3 (classified)
Support Vectors


Training SVM for laying
0 (training)
0 (classified)
6 (training)
6 (classified)
Support Vectors


Training SVM for sitting
0 (training)
0 (classified)
4 (training)
4 (classified)
Support Vectors



The accuracy of this SVM series is 57.6 percent.

NOTE: All the non linear Kernels are of Quadratic order polynomial type.

confusion =

415	**235**	**179**	1	0	1
1	182	10	1	0	0
80	54	231	0	0	0
0	0	0	308	**302**	**159**
0	0	0	**139**	217	33
0	0	0	42	13	344

In this confusion matrix, the (row,col) entry represents the (predicted, actual) members. Therefore the diagonal entries are the correct elements and non diagonal elements are inaccuracies. The highest confusion is the classification of standing as sitting (4,5). Then there is a significant wrong classification of walking up as walking (1,2). There is also error in the case of (1,3) where the walking down is classified as walking. Similarly, there is also a confusion regarding (5,6) and (5,4). This is because the difference between walking up, walking and walking down is not that distinct,

which results in a decision by the SVM that is not totally accurate. Same goes for the ambiguity between laying, sitting and standing. Moreover, this data is from the accelerometer and GPS of a mobile device, which need not produce significantly distinct data for a person who is laying and sitting, considering that both are equally motionless activities. Thus, there should be better attributed if one wishes to classify this data well.

By taking only 3 classes the covariance matrix is:

covariance =

27.5773	0.0000
0.0000	3.4287

By taking the covariance of all the classes, the covariance comes out to be:

covariance =

34.8236	0
0	2.7350

This net covariance higher than the case 1.

This is not counter intuitive. While the variance in the data sets is higher in the second case, it could be possible that the inter class variation could be much higher in the case 1 which helps in better classification of data. It could be very well possible that this higher variation is induced by the new classes introduced in section 2 BUT those new classes have much lesser inter class variance that makes it much more difficult to classify. Thus it may seem counter intuitive, but one cannot decide the accuracy estimate from the overall variance of the dataset without knowing the classes.

INCREASING THE PRINCIPAL COMPONENTS TO 100: **CODE: SVM_100PC.m**

The accuracy of this SVM series is 50%

confusion =

149	171	149	3	8	0
255	66	144	2	2	1
88	228	126	1	2	1
1	3	1	320	202	41
3	0	0	153	313	0
0	3	0	12	5	494

This has LESS accuracy than the case with the 2 Principal components. There are several reasons for this accuracy. This result is not surprising due to the following reasons:

- By increasing the number of principal components it may appear that the data is resolved into better detail. While that may be true, it is quite possible that if the data was noisy, this resolution results in introducing more errors in the actual distribution of the data thereby reducing the accuracy of classification. PCA helps in eliminating the unwanted scatter of the data in directions that contribute less to the actual value.
- Considering a high value of components results in OVERFITTING of data during the training phase. PCA helps in generalizing the SVM to fit more data. Thus if one retains all the principal components, the data may behave well with the training data but not necessarily on unseen test data.
- The final reason for this change (which is more of a technical reason) is the implementation of different matlab functions to train and test the SVM namely “fitcsvm” and “predict”. Note: On testing the same sets of data with two different functions for training SVM, there was a discrepancy in the accuracy of 2%. This is due to the differences in builtin method of implementation of SVM and convergence in functions fitcsvm and svmtrain.