For Instructor/TA Use Only

Student Name:	
Student Last Name:	
Student Andrew ID:	

Q1 (max 50)	Q2 (max 50)	Total (max 100)

24-787 Artificial Intelligence and Machine Learning for Engineering Design

Homework 5

Clustering

Naming and Folder structure: Name a root folder "andrewid_hw#". Example: lkara_hw1. In that root folder, create subfolders such as q1, q2, q3 etc. each corresponding to a question in the assignment. The files and documents related to each question should go into the corresponding subfolder.

File types: Your folders and subfolders should not contain any files except .pdf, .doc, .docx, .m, .mat, .txt .zip files. If you want to take photos of your assignment, just make sure that combine all the jpg into a single pdf file and make it clear. Unorganized and illegible files will be penalized. Take care in arranging your illustrations, written solutions, photos. If any, do not scan the your hand-written solutions in the highest resolution.

What to include: Only submit the required files that you create or modify. For programming questions, include your source code (.m and/or .mat file rather than text) in your submission. All other files, including the ones we provide as supporting functionality, are unnecessary because we already have copies of them. Unless we tell you otherwise, make sure that you only submit the file you edited/changed. For .m files, we expect to click the "run" button and everything should work (obviously, after we include the additional support files and data that was given to you with the assignment).

Readme files: Provide a readme.txt within each questions' subfolders when necessary to aid grading. This is useful when there are many files of the same type. If your folder structure and your file names are logical, you should not need a readme.txt file.

Submission: Zip the entire assignment by compressing the root folder. Example: lkara_hw1.zip. Do not use .rar**.** Make sure your entire submission file is less than 20MB when zipped. You shall use the online homework submission platform. The submission entry can be found at the top right of homework/assignment page. Make sure your submission is in .zip format, otherwise it cannot be uploaded. Feel free to re-submit your homework before deadline if you find any mistake in your previous submission. Only the last submission will be graded.

1. *(50 points)*

For this problem, you are given a dataset (*data.txt*) containing 30 data points in $\mathbb{R}^2$. An important part of clustering is determining which distance metric and similarity measure to use. Common distance metrics include:

$$\text{Euclidean distance: } d(a,b) = \|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$$

$$\text{Cosine similarity: } d(a,b) = \frac{a \cdot b}{\|a\|\|b\|}$$

Methods you will use: hierarchical single linkage, hierarchical complete linkage, hierarchical average linkage, and k-means.

(a) Write MATLAB code to implement the first three hierarchical clustering for each of the two distance metrics above. Include the full dendrograms (6 total). For visualization in these cases, you may want to try using `biograph` or `dendrogram`. Aside from this, do not use any built-in MATLAB clustering functions, such as `linkage`, `cluster`, and `clusterdata`. However, you can use these functions (for instance, `clusterdata)` to test/verify your code.

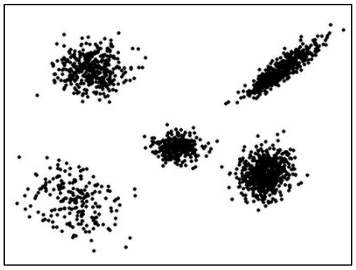
(b) For the hierarchical clusters, assume we are interested in 2 final clusters. Visualize the clusters for each of the six cases on 2D plots (make sure axes are equal). In Matlab plot, use different colors for the two clusters. Organize your plots in a table as follows:

	single linkage	complete linkage	average linkage
Euclidian distance	Plot 1,1	Plot 1,2	Plot 1,3
Cosine similarity	Plot 2,1	Plot 2,2	Plot 2,3

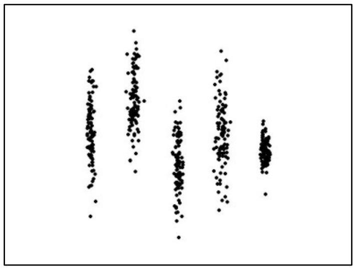
For each of the 6 cases, use the results from your hierarchical clustering algorithm to separate the data into 2 clusters. In your report, list the indices contained in each cluster. Compute the accuracy of each clustering by comparing the clusters to the ground truth provided in *labels.txt*, where each row is the true cluster of the corresponding data point in *data.txt*. Provide your code in a file `myhierarchicalclustering.m` In this file, you can implement the three methods as separate functions.

(c) Write MATLAB code to implement the k-means algorithm for each of the two distance metrics above. You may want to run your k-means with multiple random initial seeds. Do not use any built-in MATLAB clustering functions, such as `kmeans`. However, you can use this function to test/verify your code. Plot your results similar to (b), but in a 2X1 table. Report accuracy similar to the way you do in (b). Provide your implementation file in `mykmeans.m`

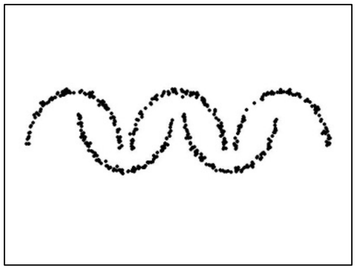
2. *(50 points)*



(a)



(b)



(c)

For each of the datasets shown above (provided as text files with this assignment), you will implement three clustering methods; you may assume that the number of clusters is known ($k = 5$). You may use Matlab's in-built functions to implement your code. Plot the results, using a different color for each cluster. A function template has been provided for you (`clustering.m`); this is the only file you need to complete for this problem.

You will use the following three methods: spectral clustering, k-means, hierarchical single link. **This problem is open ended in that you will have to study spectral clustering on your own**. Lots of information and code are readily available online.

Spectral Clustering

- Use the *normalized symmetric laplacian matrix*. You may find the following sites useful: https://en.wikipedia.org/wiki/Spectral_clustering https://charlesmartin14.wordpress.com/2012/10/09/spectral-clustering/

- Use the *gaussiandist.m* as your similarity measure for constructing the W matrix (or, could be denoted as A), from which you will also compute the D matrix. You will have to choose an appropriate value for the standard deviation in the distance function (through trial and error). With this, the normalized symmetric laplacian will be:

$$L = I - D^{-0.5} W D^{-0.5}$$

- Use Matlab's kmeans function to cluster the points obtained from the set of eigenvectors. Note that if done correctly, your first eigenvalue should be zero (rigid body translation), so you should start from the second eigenvector.

- You will have to decide the appropriate number of eigenvectors to use. Show your clustering results using the following number of eigenvectors: {1,2,5,8}. Rate the quality of the results via visual inspection. For instance:

  Quality of Spectral Clustering as a function of the number of eigenvectors:

  Data Set (a): #5 > #2 = #1 > #8

  Which means, for data Set (a), the clustering you obtain with 5 eigenvectors is better than the results for 2 eigenvectors, which is equal to the results with 1 eigenvector etc.

Produce 12 color plots for spectral clustering (3 datasets X 4 cases per dataset).

Kmeans Clustering

Cluster each of the three datasets using kmeans. Again, assume the number of clusters is 5 in all cases. You may want to explore the 'Replicates' option in Matlab's kmeans function to ensure you have multiple initializations. You can use the Euclidian distance as your distance measure.

Produce a total of 3 color plots showing the clustering results for the three data sets (one plot per data set).

Hierarchical single link

Cluster each of the three datasets using hierarchical single link. Again, assume the number of clusters is 5 in all cases. You can use the Euclidian distance as your distance measure.

Produce a total of 3 color plots showing the clustering results for the three data sets (one plot per data set).

In your report, discuss:

- For each dataset, rank the three clustering methods based on their performance. You may use the best performing eigenvector setting for spectral clustering to compare it against kmeans and hierarchical clustering. Is one method consistently better than the others? In 4-5 sentences summarize your observations. In particular, try to characterize which method is best for which kind of dataset and why.

- If labels were provided, could you use support vector machines to accurately classify these datasets? Explain your reasoning.