24-787 Artificial Intelligence and Machine Learning for Engineering Design
Homework 5
Clustering and Support Vector Machines

Unsupervised Clustering

1. *(25 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:

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

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

   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:

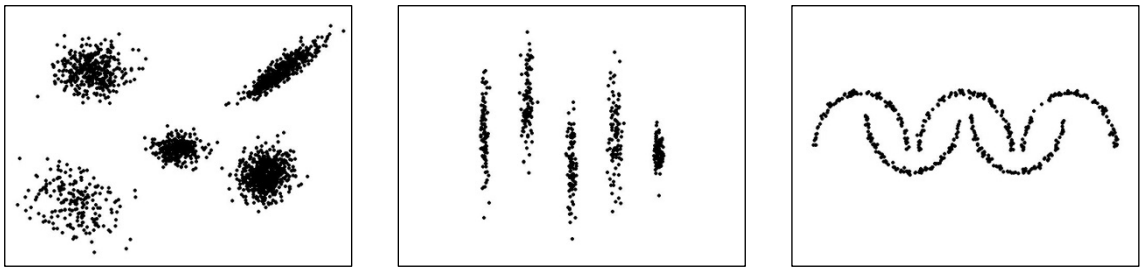
	single linkage	complete linkage	average linkage
Euclidian distance	Plot 1,1	Plot 1,2	Plot 1,3
Cosine similarity	Plot 2,1	Plot 2,2	Plot 2,3

   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. 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. *(35 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 (`hw5_clustering.m`); this is the only file you need to complete for this problem.

You will use the following three methods: k-means, hierarchical single link, and spectral clustering. 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.

Use visualization functions and table organization similar to Problem 1 to display the clustering results. You will have 9 such plots.

In your report, discuss:

- For each dataset, rank the three clustering methods based on their performance. 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.
- How can you quantitatively assess your clustering results? Consider hand-labeling the clusters (it is easy to see the meaningful clusters) and report accuracy results.
- If labels were provided, could you use support vector machines to accurately classify these datasets? Explain your reasoning.

Support Vector Machines (supervised by definition)

3. *(40 points)*

In this programming exercise, you will use support vector machines (SVM) to classify samples from a real-world dataset. Specifically, you will be predicting what activity a user is engaged in (e.g. walking, sitting, standing) based on sensor readings from their smartphone.

To get started,

- Download the Human Activity Recognition dataset from the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

- Extract all files to the directory of your choice.
- Study the dataset files. The README.txt file is a good starting point.

For this problem, you will only need to create and submit a single MATLAB script called hw5_SVM_activity_recognition.m. You may use any built-in MATLAB function.

The first thing you should do in your script is load the relevant training and testing data from the appropriate text files. There are 7352 training samples and 2947 testing samples. Each sample contains 561 features extracted from smartphone accelerometer and gyroscope timeseries data. There are 6 labeled activities (walking, walking upstairs, walking downstairs, sitting, standing, and laying).

Initially, you only need to consider samples labeled as walking, standing, or laying; ignore the other samples for now.

(a) In order to visualize the SVM decision boundary, we need to reduce the dimensionality of the data. Run principal component analysis on the data and reduce the training samples to the first two principal components. It may be helpful for you to plot the data in this reduced feature space.

(b) Since support vector machines are binary classifiers, we need a special way to handle multiple classes. Use the one-versus-all strategy on the PCA-reduced data, in which you train a separate SVM for each class (e.g. the first SVM will learn walking versus not walking). This will yield a total of three decision boundaries; include plots of the training data with the decision boundaries in your report.

(c) If an extra sample for walking was added to the training dataset that has PCA-reduced coordinates of (4,-1), which (if any) of the decision boundaries would change? Explain your reasoning.

(d) Use the SVMs learned in part (b) to classify the test data (remember, only use data for three classes right now). Use a winner-takes-all strategy, in which a sample is labeled with the class that has the highest confidence (distance to the decision boundary). Report test accuracy and show the confusion matrix.

(e) Which clustering approaches from Problem 2 would work well on the training data used here? Implement one of them and then use the cluster centers to assign labels to the test data. Report test accuracy and compare to the SVM approach.

Now you should consider all activity labels.

(f) Repeat parts (a), (b), and (d) on all the training data. You should have six trained SVMs.

(g) Looking at the confusion matrix, which classes does the computer have a difficult time distinguishing between? Does this make intuitive sense? Explain your reasoning. Think about where the data comes from.

(h) Hopefully, you noticed that the test accuracy for all six activities is much less than when only three activities were considered. Yet, the amount of variance that is explained by the first two principal components is actually higher when considering all classes (you should actually verify this). Is this counterintuitive? Why or why not?

(i) Increase the number of principal components to 100 and retrain SVMs using all activity classes. Report test accuracy and the confusion matrix. Discuss the results.