Assignment 3

SENG 474 / CSC 578D
Data Mining – Spring 2018
Totals: SEng 474: 40 ponts; CSc 578D: 55 points
Worth 5% of your final grade

Due: March 27th

Submit a pdf through ConneX before 11:55pm. I will accept submissions up to 24 hours late for a 10% reduction in your mark. After that point, no submissions will be accepted.

Show your work for all questions.

Different marking schemes will be used for undergrad (SEng 474) and grad (CSc 578D) students. Undergrad students do not have to answer the grad questions.

All code questions use the python and/or the sckit-learn library. You may install it, along with the NumPy and SciPy libraries on your own computer. Alternatively, you can work in the lab.
http://scikit-learn.org/stable/install.html
http://www.numpy.org/

1: Bisecting K-means (SEng 474, CSc 578D: 20 points)
A) Using the supplied python skeleton bisect_kmeans.py, please implement bisecting kmeans. Chose the largest cluster to split at each iteration. **Hand in your completed code**.

B) Create a 2D dataset for which bisecting k means finds the correct clustering, but scikit learn's kmeans implementation will not (even with several random restarts). Both clusterings should use the same K. **Hand in 2 plots** of the dataset, 1 showing the kmeans clustering, and 1 showing the bisecting k means clustering.

C) Create a 2D dataset for which bisecting k means cannot find the correct clustering, but scikit learn's kmeans implementation can (given several random restarts). Both clusterings should use the same K. **Hand in 2 plots** of the dataset, 1 showing the kmeans clustering, and 1 showing the bisecting k means clustering.

2. Agglomerative clustering: choosing the clusters to merge (SEng 474, CSc 578D: 10 points)
In class, a student asked if these two methods of computing cluster distance are equivalent:

i. Calculate the cluster centroids (mean point of all points in a cluster) and merge the two clusters with the closest centroids

ii. Calculate the average pairwise distance between all points across pairs of clusters, and merge those clusters with the smallest average pairwise distance.

If the two are equivalent, prove it mathematically. If the two are different, supply a (simple) case where the two measures will select different clusters to merge, give the average pairwise distances and cluster centroid distances, and show a plot of your constructed dataset.

3. HMMs (SEng 474, CSc 578D: 10 points)

Here is an HMM diagram, adapted from Speech and Language Processing (Jurafsky & Martin, 3rd ed. draft). The temperature of day (hidden state HOT or COLD) changes the number of ice cream cones Dan Jurafsky eats (1, 2, or 3). Since the day's temperature impacts the number of ice creams eaten, we can infer the hidden states from just the observed ice cream consumption.

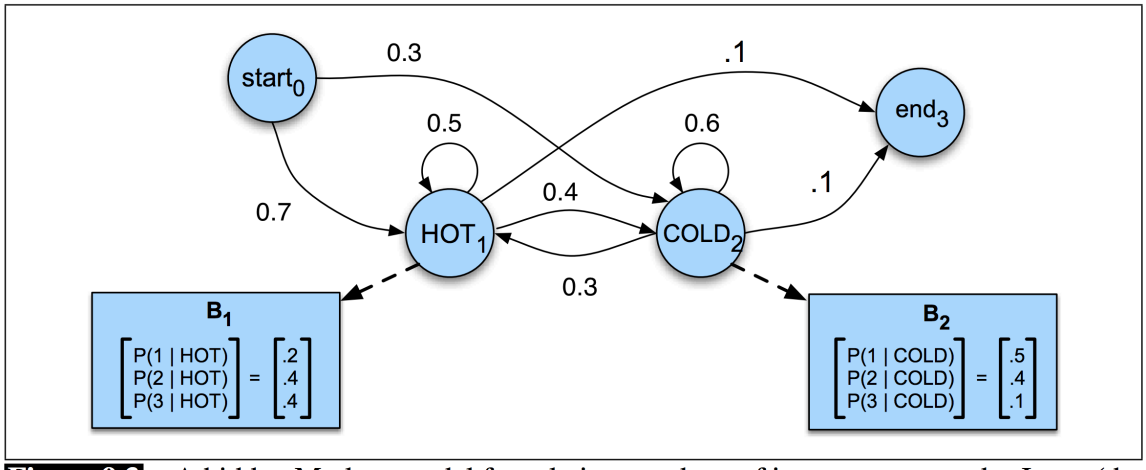

Figure 9.3 A hidden Markov model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables).

A) Give an observation sequence with a most probable hidden state sequence that starts with a HOT day and ends with a COLD day. Show your probability calculations.

B) Give an observation sequence with a most probable hidden state sequence that starts with COLD day and ends with a HOT day. Show your probability calculations.

4. Expectation Maximization (EM) (CSc 578D only: 15 points)

A) Using the supplied skeleton code in em.py, implement the EM algorithm for the case where the variance can differ between dimensions, but dimensions are assumed to be independent. This means each cluster's covariance matrix will be diagonal, but the elements along the diagonal (corresponding to the variance of each dimension) can differ. Your stopping criteria should be an *average absolute change* in cluster means of less than 0.001. **Hand in em.py including your implementation.**

B) Create a 2D dataset for which K means cannot find the correct clustering, but your EM implementation can. Both clusterings should use the same K. **Hand in 2 plots** of the dataset, 1 showing the kmeans clustering, and 1 showing the EM clustering.