Agglomerative Clustering
See Dropbox for Due Date
Thomas Kinsman

Homework is to be programmed in R, Python, Java, or Matlab. (Not Excel.)

As always, assume that the instructor has no knowledge of the language or API calls but can read comments. **Use prolific comments** before each section of code, or complicated function call to explain what the code does, and why you are using it.

Hand in your results, and the well-commented code, in the associated dropbox.

Feel free to look over each other's shoulders, at each other's work, but do you own work.
Let me know whom you worked with. Do not hand in copies of each other's code. You should be able to answer questions about your code if you see it again later.

ASSIGNMENT:
Assume you work at SSS – (Sam's Spiffy Supermarket).

SSS tracks each receipt by "Guest ID".

In order to improve their statistics on the guests, we have consolidated 10 of each guest's most recent visits into a single record for 10 purchases. This is called *marginalized data*. Notice that this is a form of data cleaning (or noise removal). Instead of averaging over 10 visits to the store, we just add up 10 of them. Remember this technique – collecting larger amounts of data is a form of noise removal or data cleaning.

A data file will be provided for you at the usual place.

SSS already knows that most of their guests are family purchases. They also have a second group of "party animals" that only buy groceries when they cannot find enough free food on campus.

Additionally, SSS suspects a third, yet unidentified, hidden group.

Your task is to identify this group, and give them a shortcut name for the marketing department to use. If they exist, we need to know how their shopping trends differ from the other two groups. What makes them special? What should we send them coupons for?

Effectively, you are identifying a third "prototype" shopper for the marketing department to pay attention to.

Prototype identification. We cluster to learn the structure of our data. In this case we have lumps in the data – or blobs. We replace hundreds of customers with a few prototypes. We use clustering to identify the prototypes. For our purposes, a prototype is a "prototypical representation". For us it is some mathematical model of the center of a cluster, and the extent of the cluster. (In other classes a prototype is an initial version of a product.)

You know two of the prototypes: family shoppers and party animals. What is the third?

The details of your assignment follow:

To simplify grading, the assignment must be very specific.

1. You are provided with the file `HW_AG_SHOPPING_CART_v???.csv`. It contains data for the number of times various categories of items (attributes) were purchased by 33 to 300 guests, for 10 different visits.

   We used 10 different visits to help get enough data on each shopper to start to draw conclusions about that shopper.

2. Compute the cross-correlation coefficients of all features with all other features.
   There exists a technique called spectral clustering, which finds attributes or features that are highly cross-correlated with each other.

   In this case I want you to perform "invariant feature rejection". Find the attribute or features that are not highly correlated with any other attribute. These features are likely to not be important.

   Find the cross-correlation coefficients of all features, and look at the table of data. Print it out in your report and remark on what you observe. Which food group(s) are highly correlated with each other? Which food group(s) is not correlated with anything?

3. Implement agglomerative clustering to cluster the guests into 3 groups as follows:
   a. At the start of Agglomerative clustering, assign each record to its own cluster prototype.
      So, you start with 100 to 300 clusters and 100 to 300 individual prototypes of those clusters.
   b. Use the Euclidean distance between cluster centers as the distance metric.
   c. Use the center of mass as the prototype center, the center of mass of a set of records, to represent its center location in data space. Use the distance between these centers as the *linkage* method.
   d. Note: At each step of clustering, only two clusters are merged together.
      As you merge, record the size of the smaller of the two clusters that are merged together.
      There are questions about this later.
   e. Cluster to completion (when you have one cluster) and answer the following questions.

   f. **Optional Bonus** 10% (1pts extra total)
      In addition to the central linkage, also implement at least two of:
      i. the single linkage,
      ii. the complete linkage,
      iii. the average linkage
      Compare and contrast the results you get.

4. Guidelines and hints: These are only hints. Your approach might be different.
   a. You need to keep track of all the records (guest id) that belong to each cluster. This is necessary because after each merge, you need to compute the new average (center of mass) of the entire cluster. This drifts after each merge.
   b. You need a separate data structure for each cluster's center of mass.
   c. It is convenient to use a data structure that records which cluster each record (guest id) is assigned to. This is the answer you are looking for ultimately when you get down to three final clusters.
   d. You may want a separate data structure that maps each original data point to its current cluster's ID to make your life easy.
   e. For big data, to be computationally efficient, you only need to re-compute the distances involved with the two clusters that are merged. For this assignment, forget about being computationally efficient – (it is hard to debug). Re-compute all the distances between all the clusters on every pass.
   f. You need some way to select the shortest inter-cluster distance, without accidentally selecting the distance from a cluster to itself. (Avoid infinite loops.)

g. It is convenient to have the lowest cluster labels persist through the progression, so that when you merge cluster 19 and cluster 29, the resulting cluster is now labeled 19. The final result is one large cluster labeled "cluster 1."
h. At each stage, you need to keep track of several things.
Update everything carefully.

Please add a section to your write-up that answers the following questions:

Questions – Copy and paste these so that you can understand the context of your answers later on:

1. You will need to remove one of the attributes in the CSV file.
Which one should you *always be certain to* remove?

2. Remark on the cross-correlation coefficients of the attributes. What information do they reveal?

3. You can keep all the other attributes, or remove some of them.
Which attributes did you finally use?

4. At each stage of clustering, you record the size of the smaller cluster being merged in.
For the last ten merges, what was the size of smaller cluster that was merged in?
What does this indicate about the true number of clusters?

5. Look at the average amount of milk, etc… purchased by the third cluster of shoppers.
What typifies the third cluster? What nick-name should we give these customers? (be polite)

6. If we switched from a "central link" to a "single link" merge step, what else would you need to add to the algorithm when computing the distance between two clusters?

7. Write a short answer question for the next midterm exam. If your question is used, you get the points on the exam. I ask this, to be sure you think about the questions that might be on the next exam.

8. Generate a dendrogram of the clusters as they are being merged.
Show the code that demonstrates your understanding of this. This is easy in R.
You can use a package in R, Python, Matlab, or Java for this, but you cannot use a web resource.
You do not need to use the same language for everything in the entire homework.

9. Bonus: (1 pt)
There is a one point bonus for implementing at least two additional linkage methods.

   Compare and contrast the different linkage methods.
   Did they make any difference for this data?
   What extra coding did you need to implement?
   Which was most difficult?
   Was the average linkage really slowest?