Peter Adam 16201859 Assignment 2 Write-up

Question 1:

KMeans has been applied to the data set and returned 2 clusters. The 1st cluster has 123 objects and the 2nd has 55.

There are 13 variables in the dataset, and the mean value of each variable in each cluster is returned. The distance between the cluster mean for each variable gives an indication of how useful that variable is for categorizing additional points into a cluster. For example, X5 and X14 have large differences between cluster means and hence would be good variables to train a classification model on.

The clustering vector returns the cluster number of each object in the original data set. 1 indicates cluster 1, 2 indicates cluster 2.

The sum of squares within each cluster gives an indication of closeness of clusters. Ratio of the sum of squares between clusters to the sum of squares within a cluster should approach 1, and the 74.2% ratio returned indicates that the clusters are fairly well separated.

Question 2:

nstart in the KMeans algorithm in R indicates how many times the algorithm is run. Since the initial positioning of the clusters is random (if the centers argument is given a number), there is the possibility that the initial cluster centers may not be optimal, and end up in local solutions that incorrectly cluster the data. As such, nstart allows the user to run the algorithm multiple times with random starting positions, in order to reduce the chance of seeing local solutions, and increase the chance of returning a stable global solution.

Question 3:

The 5 objects in the dataset have 4 attributes. To compute the dissimilarity between objects, a number of approaches need to be used together.

The first attribute has values between 0.1 and 0.6. As such, this attribute can be considered an interval-scaled variable, and the dissimilarity between objects can be calculated using a Euclidean, Manhattan or Minkowski distance.

The second object is a color, which can be considered a categorical variable. Assuming there is no ordering to these variables (blue is not closer to red than green for instance), then a ratio of mismatches like:

$$d(i,j) = \frac{p - m}{p}$$

where m is the number of matches (ie the number of variables in which i and j are in the same state), and p is the total number of variables can be computed.

The third attribute is also linear scaled, but the size of the values in this object present an issue when clustering with the small values of objects in the first attribute. As such, feature scaling could represent the two attributes on the same scale, or treating this attribute as a ratio-scaled variable and applying a logarithmic transform could also reduce the impact of the size of the data.

The final variable is Binary variable where both states appear equally valuable. As such, symmetric binary dissimilarity should be used to compute their dissimilarity, measured by:

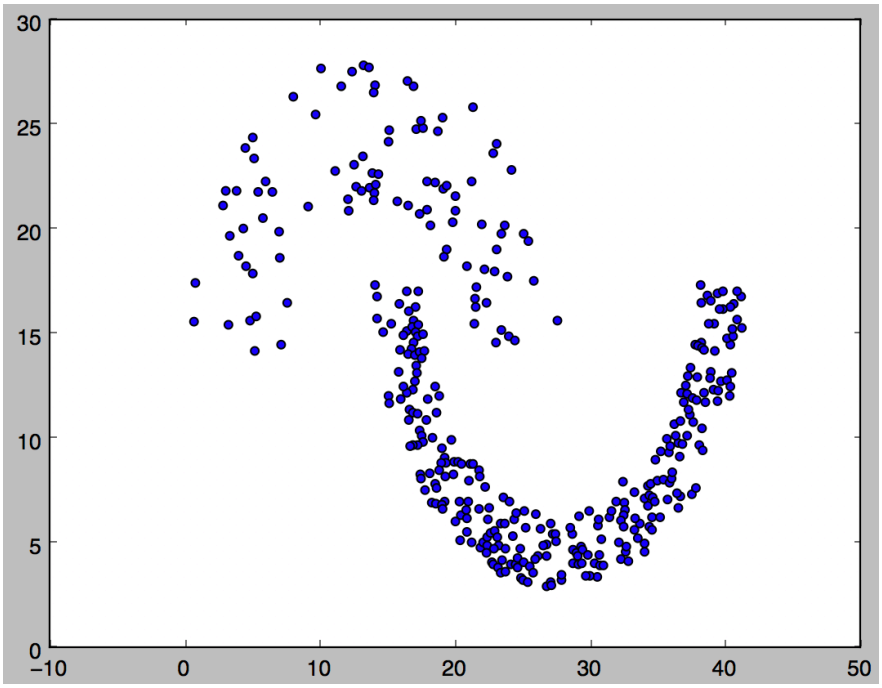
$$d(i.j) = \frac{r+s}{q+r+s+t}$$

where q is the number of variables that equal 1 for both objects i and j, r is the number of variables that equal 1 for object i and 0 for object j, s is the inverse of that and t is the number of variables the equal 0 for both objects. Total number of variables $p = q + r + s + t$.

With the dissimilarity between attributes calculated, the results can be fed into a dissimilarity matrix, upon which KMeans clustering can be performed.

Question 4:

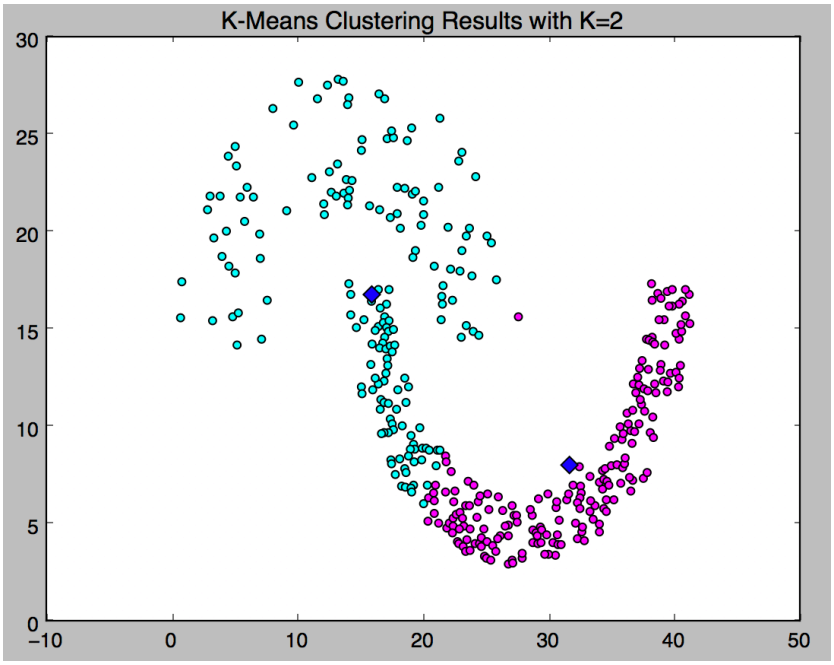
a)



b)

Code is provided in q4.py

c)



d)

KMeans failed to cluster the data into two clusters because of the shape of the clusters. The horseshoe shapes mean that when computing the distance between points and the cluster centers, there were many points in the other cluster influencing the final position of the center. KMeans using Euclidean and Manhattan distance calculations is particularly susceptible to this issue.

Clustering irregular shapes is a major challenge in clustering, and density based methods such as DBSCAN or OPTICS would be better suited to clustering this data

Question 5.

a)

This dataset contains 3900 objects each with 122 attributes. However, much of this data is redundant as it contains many empty values. As such, attributes with more than 2000 0 entries were removed. 76 columns were removed based on this criteria.

Furthermore, the data contains both anthropometric data and demographic data. Anthropometric data is most useful to a design team, while demographic data would be more useful for marketing teams. As such, we will focus only on anthropometric data.

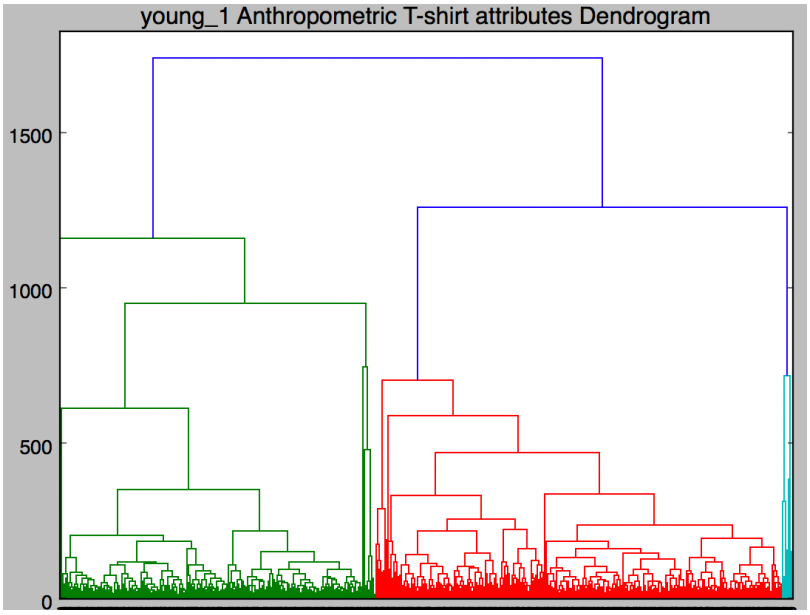
A list of 22 demographic attributes have been removed.

The data also contains a ‘Sex’ attribute. While this could be included in clustering, it would be more insightful to split the dataset on this attribute and cluster to find insights for each sex. Furthermore, ages in the dataset range from newborn to 20 years old. Due to the growth of children over this range, it may make sense to split the data up by age. The median age is 10.66, and the 75th percentile is 14. 14 seems a more reasonable age to split the data on, so 4

data subsets remain, entitled young_1, young_2, old_1, old_2 (for younger and older children, in gender 1 and 2 [the specifics of which are unknown]).

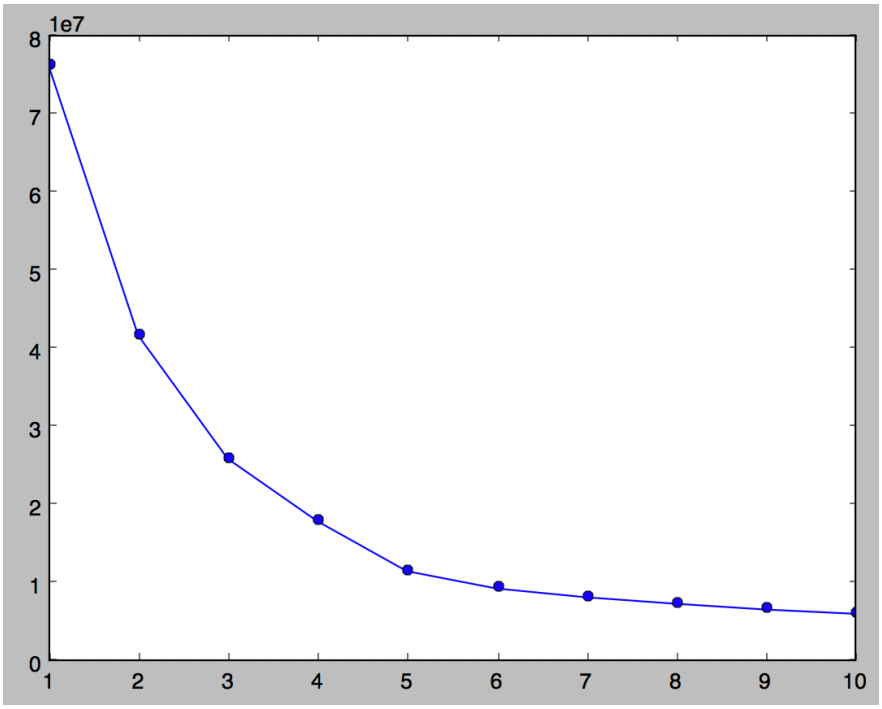
From the list of Anthropometric attributes, there are a number which would be interesting when designing a t-shirt. These include: 'CHEST CIRCUMFERENCE', 'WAIST CIRCUMFERENCE', 'SHOULDER-ELBOW LENGTH', 'ERECT SITTING HEIGHT' and 'SHOULDER BREADTH'. If designing into Small, Medium and Large shirt sizes, it would be useful to know the centroid locations of three clusters in this data.

Using the young-1 data as an example, first explore a hierarchical cluster to see if three clusters are identifiable.

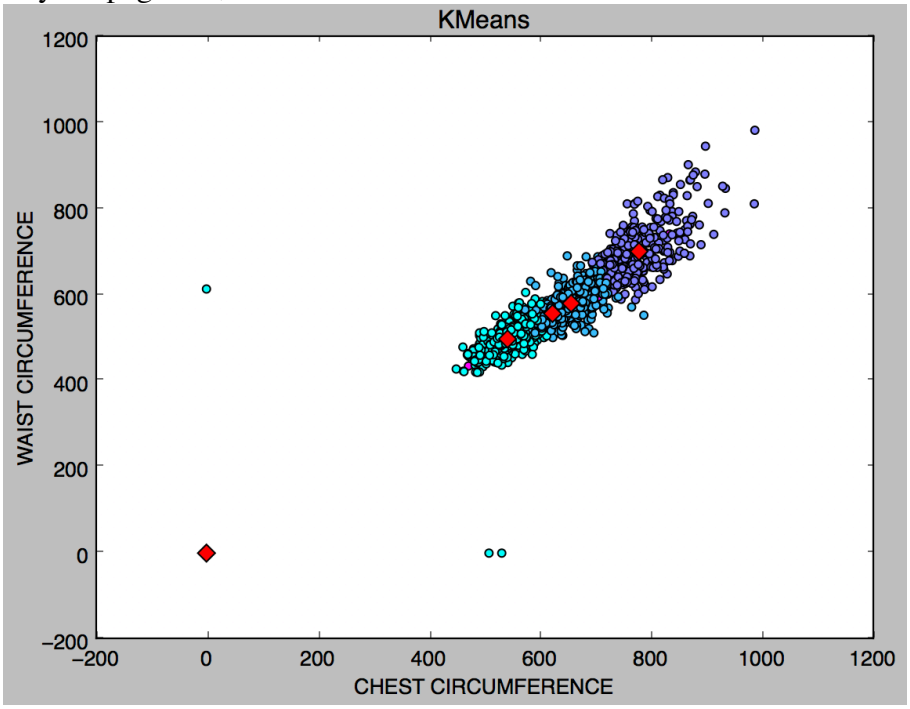


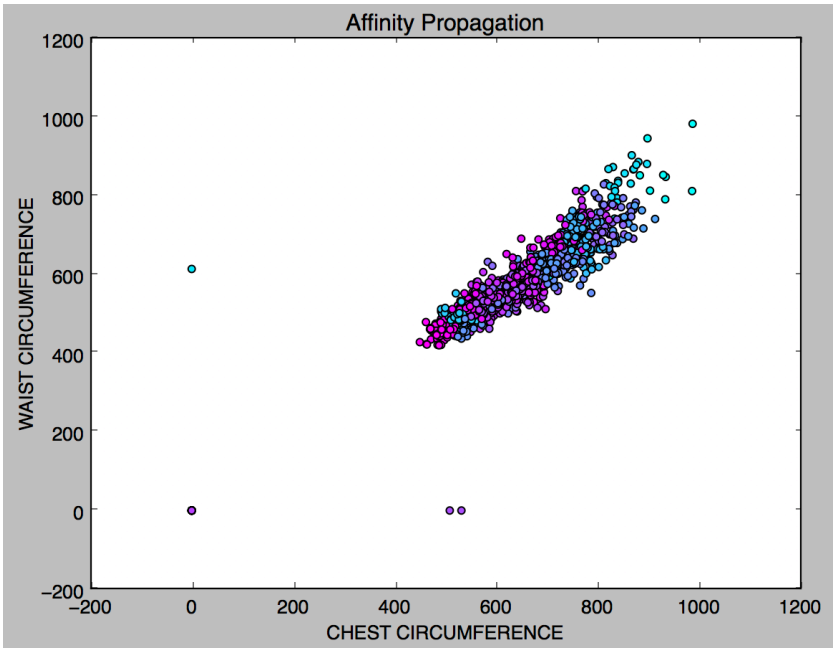
An initial hierarchical clustering suggests there are 2 major clusters within the data, but looking more closely, there are 2 small groups with data quite dissimilar to the majority at the ends of the green cluster and in the blue cluster. Ignoring this data, there appears to be 2 clusters in each of the red and blue clusters that contain a large number of objects with very similar attributes. This gives us the potential for 4 t-shirt sizes.

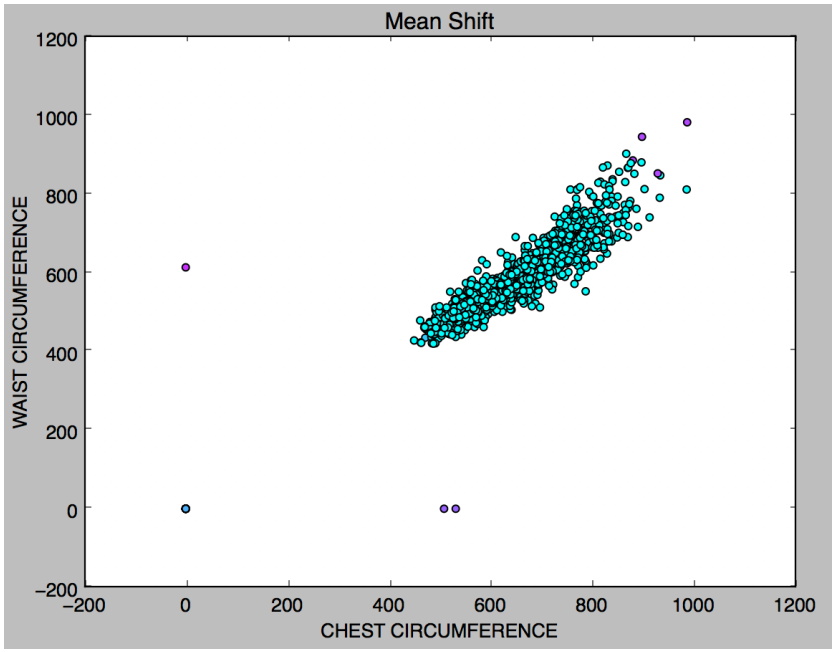
This is confirmed by plotting inertia for different numbers of clusters. The elbow of the plot (where the rate of inertia decrease begins to slow) confirms that 5 clusters seem to be present, but since we're aware that one cluster is so small that it's not worth creating a t-shirt for, we can ignore these data points at the end.



To figure out which clustering algorithm to apply, a number can be tried and the effectiveness of each measured by the inertia in the data. The algorithms applied were K-means, Affinity Propagation, and Mean Shift.







From above, the KMeans with 5 clusters appears to split the data up fairly well. There is one outlying cluster, but as we know that there will be one small cluster that is an outlier, this isn't an issue.

Breaking up the 5 clusters, we see they have the following attributes:

Cluster	Number of Objects	Chest circumference	Waist circumference	Shoulder-elbow length	Erect sitting height	Shoulder breadth
0	576	542	497	213	596	264
1	338	779	701	321	789	376
2	25	0	0	42	618	49.3
3	21	657	580	280	713	322
4	514	623	557	255	0	298

From this table, we can see that there are three clusters with significant numbers of objects, 0, 1 and 4. From looking at the centroid locations of these clusters, it would appear that cluster 0 represents the sizes of small children, cluster 4 of medium children, and cluster 1 of large children.

With these measurements, designers could have a decent idea of what dimensions to produce small, medium and large T-shirts for young children of gender 1.

A similar approach could be easily followed to predict measurements for T-shirts in the other 3 categories.

The code is contained in q5.py. The main clustering algorithm was KMeans with n_clusters=5.

With such highly dimensional data it is difficult to get a visualization of the data in two dimensions. While KMeans has an easily identifiable cluster quality metric in inertia, other clustering techniques cannot be evaluated so easily. Additionally, the high dimensionality of the data meant that there may have been many components that were not contributing large amounts of volatility to the data. Principle Component Analysis could have been performed to identify the qualities that provide the most amount of variance to the Anthropometric attributes.

The drawbacks of KMeans have already been discussed above, but it is a good 'one-size fits all' starting point, and can provides good results (like seen above). In this situation, there were erroneous clusters, but hierarchical clustering provides a great visualization to identify the impact of these clusters, and allow more specific clustering without worrying about small outlier clusters. As such, the inertia of the KMeans cluster with 5 clusters was low, and with only 50 objects (of ~1400), the 2 discarded clusters did not have an effect on the applicability of the final recommendation, but did greatly increase the accuracy of it.