Roy Chan
301202770

CMPT 459 Fall 2017
Data Mining

Assignment 1

Assignment 1.1

The K-means algorithm re-assigns values by utilizing the distance between a value and the initial centroids. Values are placed into their respective clusters based on whichever centroid the values are closer to. The resulting centroid is the mean of the clusters.

a) Let K be a 1-dimensional single data set which includes all the values of the three "natural clusters".

   K = { 1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 24, 28, 32, 36, 40}

   Using the K-means algorithm with the initial centroids of 1, 11, and 28 (assume that these centroids were chosen at random), we get the following:

K_1 = { 1 , 2, 3, 4, 5}	Resulting Centroid: 3
K_2 = { 8, 9, 10, 11, 12}	Resulting Centroid: 10
K_3 = { 24, 28, 32, 36, 40}	Resulting Centroid: 32

   Which is the set of the three "natural clusters". The algorithm detects the correct clusters.

b) Using the K-means algorithm with the initial centroids of 1, 2, and 3 (assume that these centroids were chosen at random), we get the following:

K_1 = { 1 }	Resulting Centroid: 1
K_2 = { 2 }	Resulting Centroid: 2
K_3 = { 3, 4, 5, 8, 9, 10, 11, 12, 24, 28, 32, 36, 40 }	Resulting Centroid: 17.0769

   Which does not result in set of the three "natural clusters". The algorithm does not detect the correct clusters in the first iteration, unlike a). From the results shown above, it will take many more numerations for for b) to normalize and have values be placed in their correct clusters. Poor choice of clusters will lead to more iterations. a) is one of the best case scenarios, whereas b) is one of the worst case scenarios. In a regular run of the k-means algorithm, values will be chosen at random and not pre-determined.

Assignment 1.2

a) The analogon m for the means of a cluster C for the categorical data should be chosen based on the attribute that is occurs most frequently in cluster C. The scan can construct histogram for all d attributes, therefore, m can be computable by scanning the set of objects of C only once.

b) $TD(C, m) = \sum_{p \in C} dist(p, m)$

Assume there is another object m' where m =/= m' and the following:

$$TD(C, m') < TD\ (C, m)$$

By definition and using the formula above, we get the following:

$$\sum_{i=1}^{d} \sum_{p \in C} \delta(p_i, m_i') < \sum_{i=1}^{d} \sum_{p \in C} \delta(p_i, m_i)$$

$$\sum_{p \in C} \delta(p_i, m_i') < \sum_{p \in C} \delta(p_i, m_i)$$

Which contradicts the definition, therefore the assumption cannot be possible.

c) The algorithm is extremely similar to the standard k-means algorithm with some subtle differences between the two. The algorithm represents clusters as m, whereas the standard k-means algorithm represents clusters as means. The standard k-means algorithm uses the Euclidean distance function in computations instead of a categorical distance function in our algorithm. Reassignments of objects to cluster representatives must also be checked in our algorithm because the distance function that is used can only return d different values, unlike the standard k-means algorithm.

Assignment 1.3

a) Choosing one of the clusters to split, we can split the cluster with the largest number of points where each cluster in a level have about the same number of points. If we choose to split the chosen cluster into two clusters, we can split the cluster with the largest variance so that the split clusters have individual variances. Both solutions are more or less equally as efficient as one another therefore either one could be used for our algorithm.

b) An inefficient method of splitting via the optimum split of cluster C is to enumerate all subsets of S ⊆ C. For each of the subsets, determine the distance between S and C by using one of the distance functions used in hierarchical clustering. Let the distance be D. After determining the distance, split the results into S and S-C which maximizes D. The runtime complexity of the inefficient optimal split of cluster C is $O(2^n)$ where $n = |C|$, since we have to consider every subset of C.

A more efficient method would be to initially determine the distance between all pairs of values for dist(x, y) where x, y ∈ C. Choose an arbitrary pair with the largest distance as the split of two sub clusters of C. The assign remaining values accordingly. Instead of considering every subset of C, this method examines every points of C in pairs, therefore the runtime complexity of this method is $O(n^2)$.