CMSC 691 - Intro to Data Science
Homework 5
Name: Prasad Akmar
Student ID: LE10772

10.2 Suppose that the data mining task is to cluster points (with (x, y) representing location) into three clusters, where the points are

A1(2,10), A2(2,5), A3(8,4), B1(5,8), B2(7,5), B3(6,4), C1(1,2), C2(4,9).

The distance function is Euclidean distance. Suppose initially we assign A1, B1, and C1 as the center of each cluster, respectively. Use the k-means algorithm to show only.
(a) The three cluster centers after the first round of execution.
(b) The final three clusters

Solution:

(a) The three clusters after first round of execution
(2, 10), (6, 6), (1, 2)

(b) The final three clusters
(3.66, 9.0), (7.0, 4.33), (1.5, 3.5)

10.10 Why is it that BIRCH encounters difficulties in finding clusters of arbitrary shape but OPTICS does not? Propose modifications to BIRCH to help it find clusters of arbitrary shape.

Solution:
BIRCH algorithm uses Euclidean distance and inter-cluster proximity for measuring the distance, because of which the cluster formed are perfectly spherical in shape. OPTICS algorithm on the other hand uses connectivity, which is density-based. These clusters grow by connecting points within a defined radius. This can lead in finding more accurate arbitrary-shaped clusters.

We could modify BIRCH to use a density-based and connectivity-based distance measure instead of proximity-based to cluster low-level trees and build the levels of the tree. This would lead to creation of connectivity-based arbitrary-shaped clusters.

10.12 Present conditions under which density-based clustering is more suitable than partitioning-based clustering and hierarchical clustering. Give application examples to support your argument.

Solution:
Partitioning-based and hierarchical clustering work well with data which is evenly distributed in spherical or circular shapes. These algorithms are not as good in filtering out noise/outliers, as they try to include every point in one of the clusters.

In density-based clustering, clusters have more accurate boundaries. So, they can determine arbitrary shaped clusters whereas algorithms based on distance can only determine spherical shaped clusters. This method can easily recognize noise as well as outliers. Hence often used in determining outliers or filtering out noise.

Example: Credit card fraud detection and the monitoring of criminal activities in electronic commerce can be done very efficiently, because of the accuracy in cluster definition and therefore outlier detection.

10.18 Suppose that you are to allocate a number of automatic teller machines (ATMs) in a given region so as to satisfy a number of constraints. Households or workplaces may be clustered so that typically one ATM is assigned per cluster. The clustering, however, may be constrained by two factors:
(1) obstacle objects (i.e., there are bridges, rivers, and highways that can affect ATM accessibility), and
(2) additional user-specified constraints such as that each ATM should serve at least 10,000 households. How can a clustering algorithm such as k-means be modified for quality clustering under both constraints?
For this problem I'm just looking for you to propose some ideas. In addition to what's asked for in the book, say something about how your proposed modifications impact the computational complexity of the clustering algorithm.

Solution:
k-means can be modified in the following way:
To satisfy the first constraint, i.e. obstacle objects, we could treat the regions divided by obstacle objects as different subproblems and run K-means individually on each one of them. In this way, we don’t ever end up overlapping an area in two clusters.

To satisfy the second constraint, i.e. to ensure that one ATM serves to at least 10000 households/workplaces, we can put a check on the formed clusters in each subproblem if they contain more than 10000 nodes and assign an ATM in that cluster only if there exist more than 10000 nodes. Density-based clustering uses similar approach to ensure a minimum number of points in a cluster.

Performance wise, k-means will be having little or no impact as we are not making any modifications to the algorithm itself, but the whole process will be computationally expensive and slow as we are running multiple instances of k-means on subproblems and trying to satisfy both constraints, second of could significantly slow down the formation of clusters.
Computational complexity of the resultant algorithm will be impacted as the value of k in O(n^{dk+1}) *(d=dimension, k=number of clusters)* cannot be predicted at the start, but it will not go beyond a maximum value. So, no significant change.