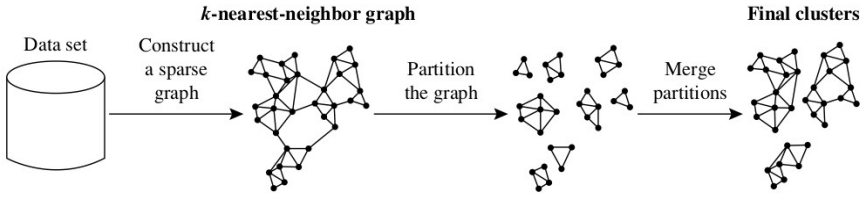
HW1-REPORT

Can Duyar - 171044075

* Source of the 2D data sets that I used = https://elki-project.github.io/datasets/



(Chameleon clustering technique explained in book. I used these steps to implement it)

Phase 1: It uses a graph partitioning algorithm to divide the data set into a set of individual clusters.
Phase 2: it uses an agglomerative hierarchical mining algorithm to merge the clusters.

I used various data sets to implement the chameleon algorithm. Chameleon is a hierarchical clustering algorithm that uses dynamic modeling to decide the similarity among pairs of clusters. Chameleon needs the k-nearest-neighbor graph technique to make a sparse graph, where each vertex of the graph defines a data object, and there exists an edge among two vertices (objects) if one object is between the k-most-similar objects of the other. The edges are weighted to reflect the similarity among objects. Chameleon uses a graph partitioning algorithm to partition the k-nearest-neighbor graph into a large number of relatively small subclusters.

Showing extracted clusters for at least 3 values of each parameter:

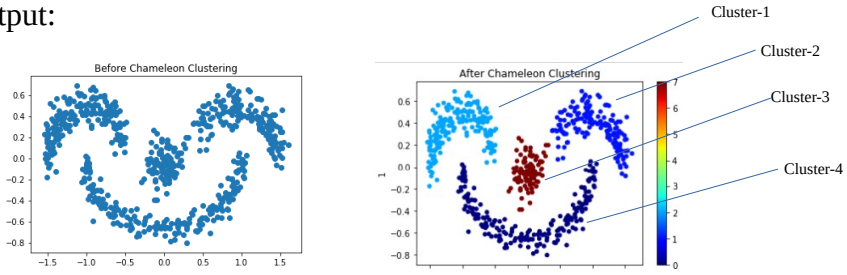
Effect of “number_of_clusters” parameter on output:

1) for knn_value = 20 number_of_clusters = 4

```
knn_value = 20
number_of_clusters = 4

chameleonClustering(dataFrame,number_of_clusters,knn_value)
```

output:

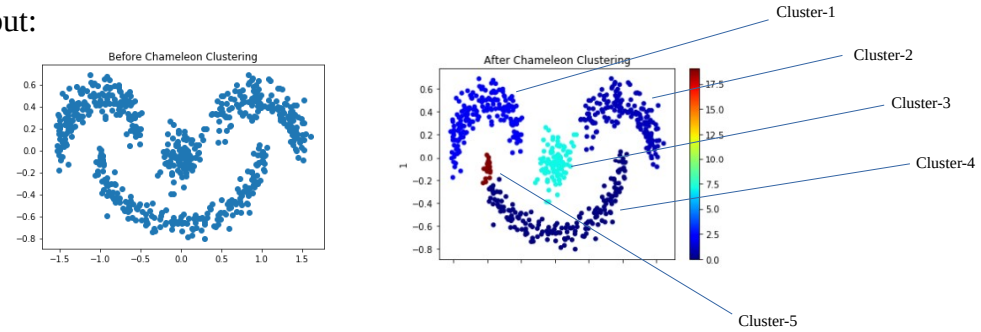


2) for knn_value = 20 number_of_clusters = 5

```
knn_value = 20
number_of_clusters = 5

chameleonClustering(dataFrame,number_of_clusters,knn_value)
```

output:

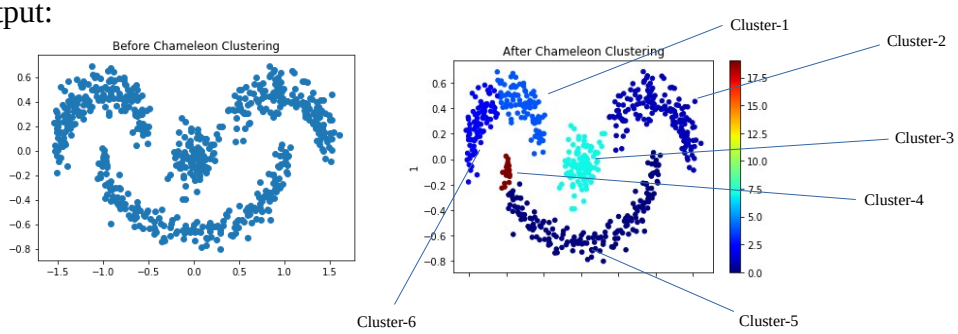


3) for knn_value = 20 number_of_clusters = 6

```
knn_value = 20
number_of_clusters = 6

chameleonClustering(dataFrame,number_of_clusters,knn_value)
```

output:



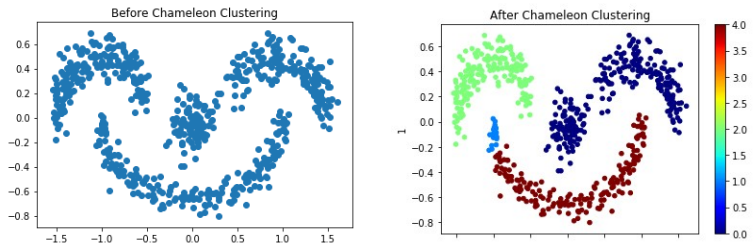
Effect of “knn_value” parameter on output:

1) for knn_value = 5 number_of_clusters = 4

```
knn_value = 5
number_of_clusters = 4

chameleonClustering(dataFrame,number_of_clusters,knn_value)
```

output:

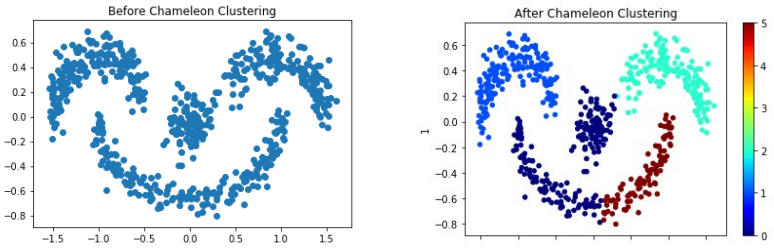


2) for knn_value = 10 number_of_clusters = 4

```
knn_value = 10
number_of_clusters = 4

chameleonClustering(dataFrame,number_of_clusters,knn_value)
```

output:

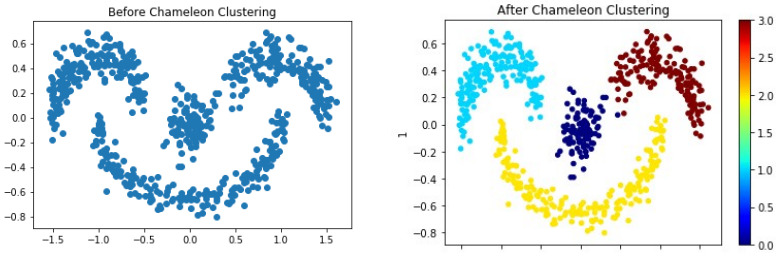


3) for knn_value = 15 number_of_clusters = 4

```
knn_value = 15
number_of_clusters = 4

chameleonClustering(dataFrame,number_of_clusters,knn_value)
```

output:



How the parameters effect the results?

If I change my “number_of_clusters” parameter then I decide how many clusters there will be. For example, I gave 4 as “number_of_clusters” in my first example and I had 4 clusters etc. If I change “knn_value” parameter then it determines the closest number of points(with the help of Euclidean distance), which significantly affects clustering (as far as I see from my outputs).

What are the advantages and disadvantages of the algorithm? Write a discussion about it while comparing it with other clustering techniques:

Advantages:

-Chameleon can discover natural shapes of different shapes and sizes
-Merging decision dynamically adapts to the different clustering model characterized by the clusters in consideration.
-The methodology of dynamic modeling of clusters in agglomerative hierarchical methods is applicable to all types of data.
-It uses dynamic model because it considers the internal characteristics of the clusters themselves.
-This algorithm is proven to find clusters of diverse shapes, densities, and sizes in two-dimensional space.

Disadvantages:

-CHAMELEON is known for low dimensional spaces, and was not applied to high dimensions
-Chameleon has been shown to have greater power at discovering arbitrarily shaped clusters of high quality than several well-known algorithms such as BIRCH and density based DBSCAN. However, the processing cost for high-dimensional data may require O(n^2) time for n objects in the worst case.

What is the time complexity of chameleon. How it is comparing to the other clustering techniques?

For large n, the worst-case time complexity of the algorithm is O(n(log2 n + m)), where m is the number of clusters formed after completion of the first phase of the algorithm Time complexity of CHAMELEON algorithm in high dimensions is O(n^2). It is more costly than other models as it will cost more time in high-dimensional data.

Method	Algorithm	Scalability	Cluster shape	Outliers sensitivity	Time complexity
Hierarchical	BIRCH	Active	Spherical	No	O(n)
	CURE	Active	Arbitrary	No	$O(n^2 \log n)$
	ROCK	Moderate	Arbitrary	No	$O(n^2 \log n)$
	Chameleon	Active	Arbitrary	No	$O(n^2)$
Partitioning	K-means	Moderate	Spherical	Yes	O(n k d)
	K-medoids	Passive	Spherical	No	$O(k(n-k)^2)$
Density-based	DBSCAN	Moderate	Arbitrary	No	O(n log n)
	OPTICS	Moderate	Arbitrary	No	O(n log n)
Grid-based	STING	Active	Arbitrary	No	O(n)
	CLIQUE	Active	Spherical	Yes	$O(n+d^2)$