Data Mining - Assignment 1

Djuro Radusinovic - 171044095

November 29, 2021

1 Introduction

Chameleon find the clusters in the data set by using two phase algorithms.
It first uses a graph partitioning algorithm in order to produce large number of small sub-clusters.
Second, it uses an agglomerative hierarchical algorithm to find the genuine clusters by repeatedly combining this smaller sub-clusters.

In order to perform these steps it takes the data set and makes a sparse graph using K-nearest neighbourhood graph. After that it partitions this graph in order to make many sub-clusters. When sub-clusters are formed they are merged with respect to parameters (inter-connectivity and closeness) of this algorithm and after partitions are properly merged final clusters are formed. Also edge weight of the dense regions are very high and in sparse regions they tend to be low.

2 Advantages and disadvantages of the algorithm

2.1 Advantages

One of the most important advantages of Chameleon clustering algorithm is that it can produce any shaping with its clustering. This way it takes into account special characteristics of individual clusters. It is a dynamic clustering algorithm so it overcomes static clustering algorithms limitations. It also overcomes cluster similarity constraint since it merges clusters according to their inter-connectivity and closeness. It is applicable to all types of data.

Another advantage would be that we are here using k-nearest neighbour graph.
This way data points that are far in the graph will be disconnected.
It makes neighborhood completely dynamically.
Density of the region is recorded as weights of the edges.
It also has a computational advantage compared to full graph in many algorithms that operate on graphs.

2.2 Disadvantages

This algorithm won't handle noise at all so data has to be preprocessed and cleaned so that we won't have inconveniences.
It also doesn't handle singleton/stand-alone clusters.
It is not very fast and should be used for specific cases.

3 Parameters of Chameleon clustering algorithm

3.1 K

K-nearest neighbour graph
Represents the number K number of nearest vertices it is connected to.

3.2 MINSIZE

The minimum size of the initial cluster. When performing partitioning initially all points belong to the same cluster. After that we continue performing partitioning until **size of all clusers is smaller than MINSIZE parameter**. This parameter is used in hMETIS. When using hMETIS we split the graph into two sub-clusters the edge-cut between clusters is minimized. We select the largest cluster among our sub-clusters and we bisect it. We perform this in a loop until the larger sub-cluster contains fewer than a specified number of vertices (MINSIZE *should be between 1 - 5 percent*).

3.3 $\mathbf{T}_{ri}$

Threshold of related inter-connectivity
$if RI(C_i, C_j) \geq T_{ri}$ than the cluster's inter-connectivity is good enough for merging.

3.4 $\mathbf{T}_{rc}$

Threshold of related intra-connectivity
$if RC(C_i, C_j) \geq T_{rc}$ than the cluster's clonesness is good enough for merging.

3.5 α

Coefficient for weight of RI and RC.
if $\alpha > 1$ -> Chameleon algorithm will give higher importance to the RI (relative closeness)
if $\alpha < 1$ -> Chameleon algorithm will give higher importance to the RC (inter-connectivity)

4 Time complexity

Here we have two parameters **N**(number of item data items) and **M**(number of initial sub-clusters produced by the graph partitioning algorithm).
Each initial sub-cluster will have the same number of nodes of N/M.
First, we need the k-nearest neighbor graph.
For a low-dimensional data set it would execute in $\mathbf{O}(NlogN)$
But for the high-dimensional data set it would need $\mathbf{O}(N^2)$
For graph partitioning algorithm we will need $\mathbf{O}(|V| + |E|)$
Of course, as we are using k-nearest neighbor graph $|E| = O(|V|)$

Now to bisect each initial cluster, its worst case would be $\mathbf{O}(N/M)$**, making it O(N) for worst case.**
For merging we would than need **O(NM)**
To find the most similar pair of clusters we need $\mathbf{O}(M^2logM)$
This gives the final time complexity of $\mathbf{O}(NM + NlogN + M^2logM)$

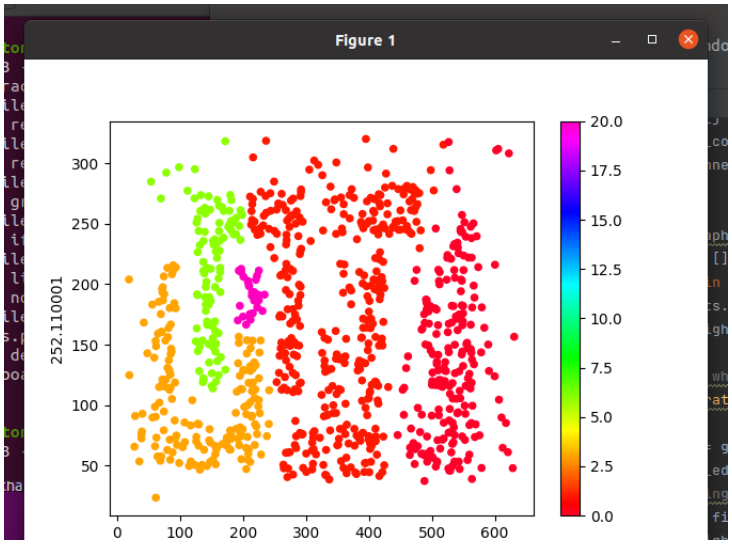

Figure 1: K=5 $K_N = 20M = 40a = 0.7$.

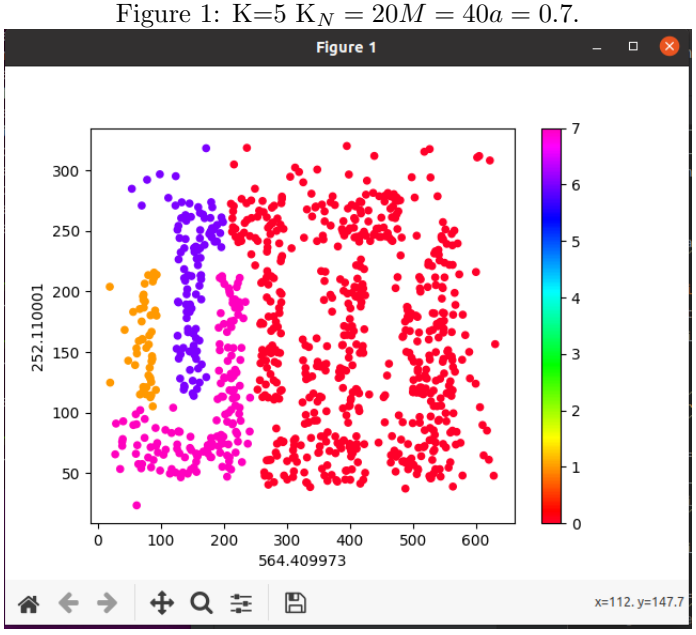

Figure 2: K=5 $K_N = 25M = 50a = 1.0$

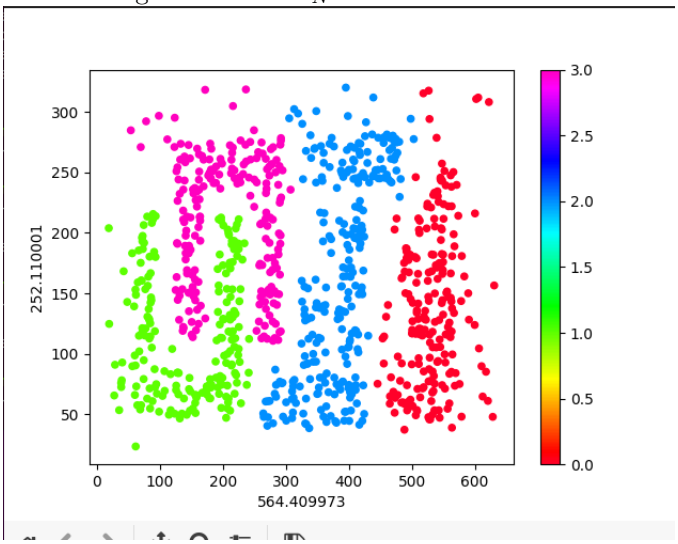

Figure 3: K=4 $K_M = 10M = 20a = 2.0$.