Implementation of Efficient Datasets Using Different Clustering Techniques

Harish G R¹, Muralidharan P², Dinesh M³, R.Janarthanan⁴, Badi Alekhya⁵
B.E. Scholar, Department of CSE, T.J.S Engineering College, Chennai, India¹²³, R.Janarthanan⁴
Assistant Professor, Department of CSE, T.J.S Engineering College, Chennai, India¹²³.
Professor, Department of CSE, T.J.S Engineering College, Chennai, India⁵

ABSTRACT: In data mining, Clustering is the most popular, powerful and commonly used unsupervised learning technique. It is a way of locating similar data objects into clusters based on some similarity. Clustering algorithms can be categorized into seven groups, namely Hierarchical clustering algorithm, Density-based clustering algorithm, Partitioning clustering algorithm, Graph-based algorithm, Grid-based algorithm, Model-based clustering algorithm and Combinational clustering algorithm. These clustering algorithms give different result according to the conditions. Some clustering techniques are better for large data set and some gives good result for finding cluster with arbitrary shapes. This paper is planned to learn and relates various data mining clustering algorithms. Algorithms which are under exploration as follows: DBSCAN, OPTICS, FOPTICS, HDBSCAN, GDBSCAN, DeLi-Clu, DiSH, HiSC, HiCo and DENCLUE. This paper compared all these clustering algorithms according to the many factors. After comparison of these clustering algorithms I describe that which clustering algorithms should be used in different conditions for getting the best result.

I. INTRODUCTION

Clustering is most typically used and a lot of powerful unsupervised learning technique in data processing. It is helpful method that aims to arrange the input data set in to a collection of finite range of semantically consistent group with respect to some similarity. These algorithms will be roughly classified into seven classes, particularly Hierarchical algorithms, Density-based algorithms, Partitional algorithms, Graph-based algorithms, Grid-based algorithms, and Model-based algorithms. Among these varieties of algorithms, Density-based algorithms are measure renowned for his or her easy explanation and therefore the relative straightforward implementation. Another two vital benefits of this algorithms are measure it is able of discovering clusters of various shapes and different size even in outlier data set and it does not need users to specify the amount of clusters. The purposed Density-based algorithms are distinguishing dense regions that are measure separated by low-density regions.

II. RELATED WORKS

The DBSCAN (Density Based Spatial Clustering of Application with Noise) [2] is the basic clustering algorithm to mine the clusters based on objects density. In this algorithm, first the number of objects present within the neighbour region (Eps) is computed. If the neighbour objects count is below the given threshold value, the object will be marked as NOISE. Otherwise the new cluster will be formed from the core object by finding the group of density connected objects that are maximal w.r.t density reachability. The cluster formed by the DBSCAN algorithm will have wide variation inside each cluster in terms of density.

The OPTICS(Ordering Points To Identify The Clustering Structure) [5] algorithm adopts the original DBSCAN algorithm to deal with variance density clusters. This algorithm computes an ordering of the objects based on the reachability distance for representing the intrinsic hierarchical clustering structure. The Valleys in the plot indicate the clusters. But the input parameters ξ is critical for identifying the valleys as ξ clusters.
The DENCLUE [3] algorithm uses kernel density estimation. The result of density function gives the local density maxima value and this local density value is used to form the clusters. If the local density value is very small, the objects of clusters will be discarded as NOISE.

DeLi-Clu,[6] Density-Link-Clustering combines ideas from single-linkage clustering and OPTICS, eliminating the epsilon parameter and offering performance improvements over OPTICS.

HiSC[11] is a hierarchical subspace clustering (axis-parallel) method based on OPTICS.

HiCO[11] is a hierarchical correlation clustering algorithm based on OPTICS.

DiSH[7] is an improvement over HiSC that can find more complex hierarchies.

FOPTICS[8] is a faster implementation using random projections.

HDBSCAN* [9] is based on a refinement of DBSCAN, excluding border-points from the clusters and thus following more strictly the basic definition of density-levels by Hartigan.[11]

**DBSCAN ALGORITHM**

1. Require: Dataset={y1,y2,…yn} the dataset. r: the maximum radius of a neighborhood. MinPts: the minimum number of data degree contained in r-neighborhood.
2. Ensure: CS = {CS1, CS2,… CSk}: set of clusters.
3. ClusterID1=0
4. Tick all degrees yi belong to Dataset as “UNVISITED”
5. for all yi belong to Dataset do
6. if yi is ticked as “UNVISITED” then
7. mark yi as “VISITED”
8. count the degrees in the r-neighborhood of yi as |Nr(yi)|
9. if |Nr(yi)| < MinPts then
10. tick all degrees in the r-neighborhood of yi as “outliers”
11. else
12. ClusterID1++
13. ticked all degrees in the r-neighborhood of yi as ClusterID1
14. insert degree ticked as “UNVISITED” in the r-neighborhood of yi in to a queue
15. while the queue is not blank do
16. select the head degree yp in the queue
17. tick yp as “VISITED”
18. calculate the degree in the r-neighborhood of yp as |Nr(yp)|
19. remove yp from the queue
20. if |Nr(yp)|>=MinPts then
21. tick all degrees in the r-neighborhood of yp as ClusterID1
22. insert degrees ticked as “UNVISITED” in the r-neighborhood of yp in to the queue
23. endif
24. endwhile
Limitations of DBSCAN:

- The performance of the algorithmic program depends on two parameters, r and MinPts
- The time consumption for looking the closest neighbors of every object is intolerable within the cluster enlargement.
- Selecting completely different beginning degrees leads to quite different consequences.
- DBSCAN is unable to spot adjacent cluster therefore numerous densities.

OPTICS ALGORITHM

OPTICS (SetOfObjects, e, MinPts, OrderedFile)
OrderedFile.open();
FOR i FROM 1 TO SetOfObjects.size DO
Object := SetOfObjects.get(i);
IF NOT Object.Processed THEN
ExpandClusterOrder(SetOfObjects, Object, e, MinPts, OrderedFile)
OrderedFile.close();
END; // OPTICS
DeLiClu( SetOfObjects S) 
kNNJoin(S,S); 
copy the index storing S to the index storing R; 
s := start object 2 S; 
write (s, 1) to output; 
migrate s from S to R; 
add pair (S.root,R.root) to priority queue; 
while S 6= ; do 
p:= minimum pair in priority queue; 
if p = (nS, nR) is a pair of nodes then 
insert all combinations of (nS.children, nr.children) into priority queue; 
else p = (s, r) is a pair of objects 
write (s, denDist(s, r)) to output; 
reInsertExpanded(path(s), root); 

Fig-3: Output of DeLiClu Algorithm

HDBSCAN ALGORITHM:

1. ClusterID1= 1
2. Tick all degrees yi belong to Dataset as “UNCLASSIFIED”
3. calculate the influence space ISk (yi) of each degree yi belong Dataset
4. for all yi belong to Dataset do 
5. if yi is ticked as “UNCLASSIFIED” then 
6. if ExpandCoreCluster(yi,ClusterID1) then 
7. ClusterID1++
8. end if 
9. end if 
10. end for 
11. for all yi belong to Dataset do 
12. if yi is ticked as “UNCLASSIFIED” then 
13. search each degree yj belong to ISk(yi) in the influence space of 
14. end if 
15. end for 
16. end function 
17. function EXPANDCORECLUSTER1 (yi, ClusterID1) 
18. SeedList1=ISk(yi) 
19. if |SeedList1|>2k/3 then 
20. yi is labeled as ClusterID1
21. else
22. false
23. end if
24. for all y\textsubscript{j} belong to SeedList1 do
25. if |IS\textsubscript{k}(y\textsubscript{j})| > 2k/3 then
26. y\textsubscript{j} is labeled as ClusterID1
27. for all zm belong to |IS\textsubscript{k}(y\textsubscript{j})| do
28. if zm is labeled as “UNCLASSIFIED”
29. If zm is not in SeedList1 then
30. add zm into SeedList1
31. end if
32. end if
33. end for
34. end if
35. end for
36. true
37. end function

Fig-4: OUTPUT OF HDBSCAN ALGORITHM

DiSH Algorithm

```
algorithm DISH(D, p, e)
  w — Cluster order; // initially empty
  py — empty priority queue ordered by ReachDmr
  foreach p \in D do
    p.g = \infty;
    Compute w(p);
    Insert p into py;
  while (py \neq \emptyset) do
    q = py.next();
    r — p-nearest neighbor of w.r.t. SDist;
    foreach p \in pq do
      new.x = \max(SDist(o, r), SDist(o, p));
      py.update(p, new.x);
      append q to w;
  return w;
```

Fig-5: DiSH Algorithm
Each clustering algorithm has its own advantages and disadvantages based on given inputs. In this paper we have discussed about some clustering algorithms and performed a comparative study among those to explain the similarities and dissimilarities to know which algorithm suits best for a particular data. However every algorithm is proved to be different to that of others in certain aspects.
REFERENCES


[3]. DENCLUE-IM: A New Approach for Big Data Clustering Hajar REHIOUI, Abdellah IDRISI, Manar ABOUREZQ, Faouzia ZEGRARI https://ac.els-cdn.com/S1877050916315824/1-s2.0-S1877050916315824-main.pdf?_tid=cfa3c7e0-f613-11e7-bb91-00000aab0f26&acdnat=1515595240_6440caf40b07a25805c25bfeebe2471f


[5]. OPTICS: Ordering Points To Identify the Clustering Structure Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, Jörg Sander http://www.dbs.informatik.uni-muenchen.de/Publikationen/Papers/OPTICS.pdf

[6]. DeLiClu: Boosting Robustness, Completeness, Usability, and Efficiency of Hierarchical Clustering by a Closest Pair Ranking Elke Achtert, Christian Bohm, and Peer Kroger http://www.dbs.informatik.uni-muenchen.de/~achtert/papers/PAKDD06_delici.pdf


[10]. https://en.wikipedia.org/wiki/DBSCAN