ISSN 2321 3361 © 2018 IJESC

Research Article Volume 8 Issue No.8

Comparative study of CURE, Improved CURE and CURE-NS for Large Databases

Anand Gandhe[1], Dr. D.M. Puntambekar[2]
Associate Professor[1], Director[2]
School of Computers, IPS Academy, India

Abstract:
Data mining is a method of mining and extracting useful information from large datasets. Clustering is a process of grouping objects and data into groups or clusters so that objects in the same group are more similar to each other than to those in other groups or clusters. CURE employs a fixed number of representative points to describe the cluster, and the set of representative points are firstly chosen randomly, and then are shrunk toward the mean of cluster. Improved CURE uses appropriate linkage functions for datasets. CURE-NS uses the difference of density values of the representative points to determine the direction and distance of shrinking. Therefore in this paper we compare these three major hierarchical clustering algorithms CURE, Improved CURE, and CURE-NS according to their parameters.

Keywords: Clustering, CURE, Improved CURE and CURE-NS.

1 INTRODUCTION

Data mining is the process of knowledge discovery in databases. The goal of data mining is to extract information from a dataset and transform the information into a comprehensible structure for further use. Clustering is the unsupervised classification of data into groups or clusters. The input for a system of cluster analysis is a set of samples and a measure of similarity (or dissimilarity) between two samples. The output from cluster analysis is a number of groups or clusters that form a partition, or a structure of partitions, of the data set. Clustering algorithms can be classified into Partition, Hierarchical, Density based and Grid based. Among all we discuss the CURE (Clustering using Representative points).
This paper is aimed to have comparative study of CURE, Improved CURE and CURE-NS (New Shrinkage Scheme). All the three methods are discussed along with their algorithms, strength and limitations.

Hierarchical clustering algorithms: These schemes are further divided into –

- **Divisive algorithms (top-down, splitting):** These algorithms act in the opposite direction; that is, they produce a sequence of clustering of increasing *m* at each step. The clustering produced at each step results from the previous one by splitting a single cluster into two.

- **Agglomerative algorithms (bottom-up, merging):** These algorithms produce a sequence of clustering of decreasing number of clusters, *m*, at each step. The clustering produced at each step results from the previous one by merging two clusters into one.

2 CURE (Clustering Using REpresentatives)

In this section, we present CURE's agglomerative hierarchical clustering algorithm. It first partitions the random sample and partially clusters the data points in each partition. After eliminating outliers, the pre clustered data in each partition is then clustered in a final pass to generate the final clusters.

The following steps will involve in clustering using CURE as shown in the figure.

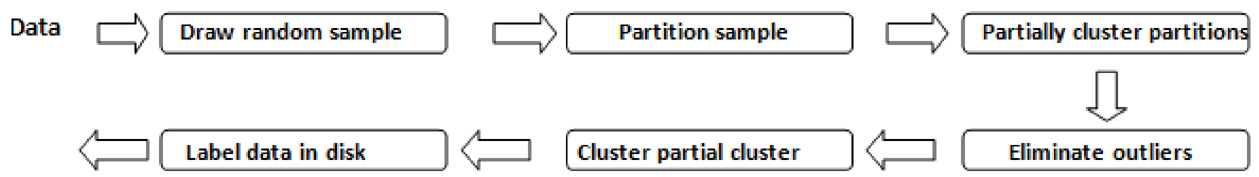


Fig1. CURE Process

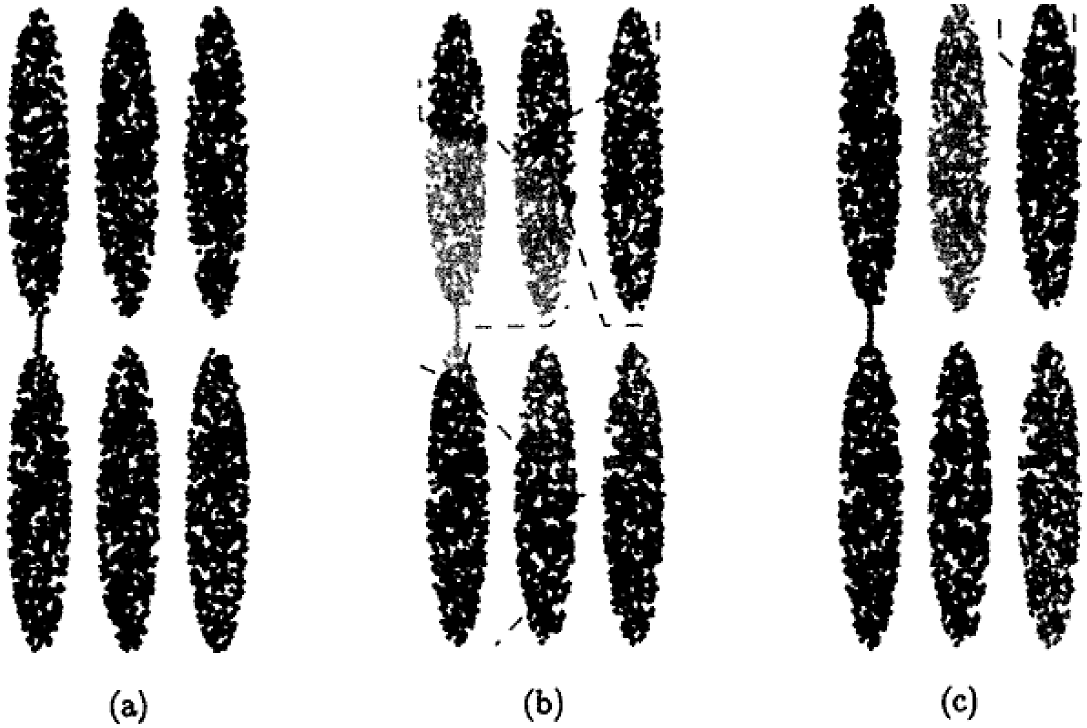


Fig 2. Clusters generated by CURE Clustering Algorithms

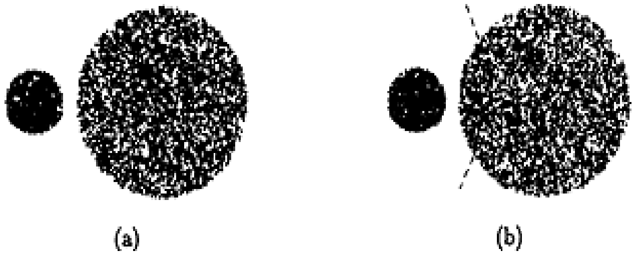
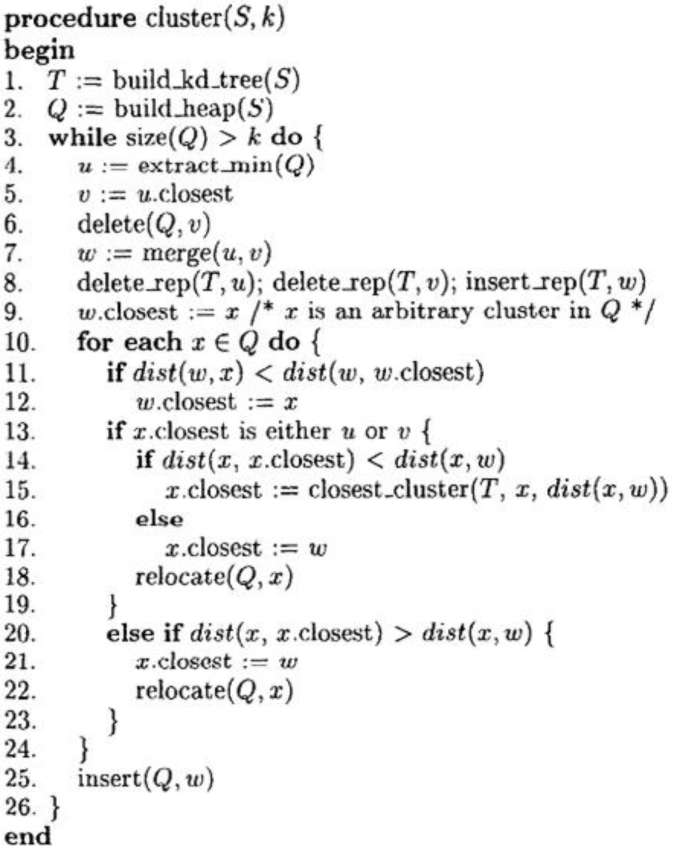
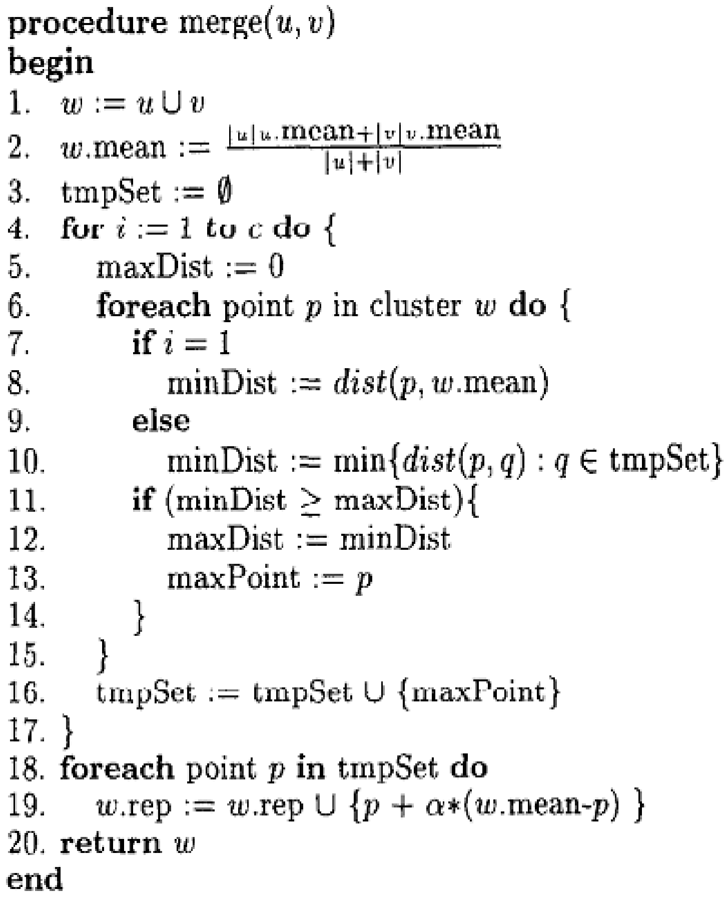


Fig 3. Problem of Labeling



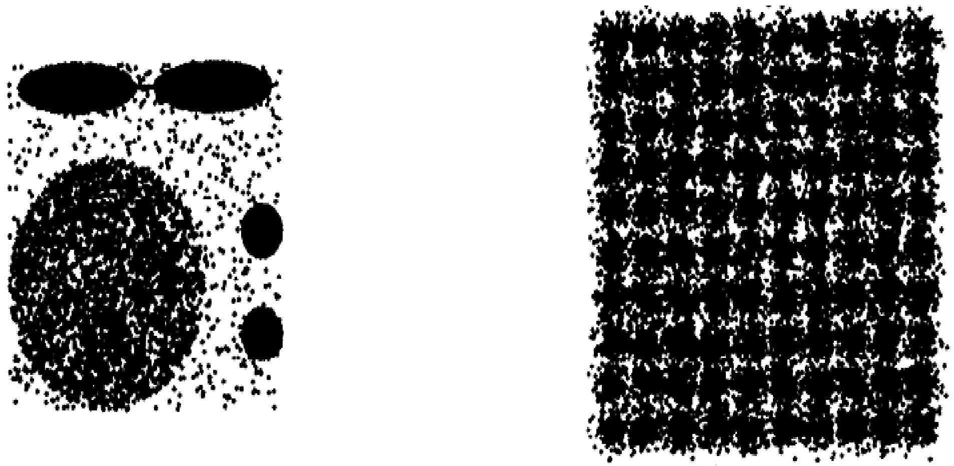
```
procedure cluster($S, k$)
begin
T := build_kd_tree($S$)
Q := build_heap($S$)
while size($Q$) $>$ $k$ do {
    $u$ := extract_min($Q$)
    $v$ := $u$.closest
    delete($Q, v$)
    $w$ := merge($u, v$)
    delete_rep($T, u$); delete_rep($T, v$); insert_rep($T, w$)
    $w$.closest := $x$ /* $x$ is an arbitrary cluster in $Q$ */
    for each $x \in Q$ do {
        if $dist(w, x) < dist(w, w.\text{closest})$
            $w$.closest := $x$
        if $x$.closest is either $u$ or $v$ {
            if $dist(x, x.\text{closest}) < dist(x, w)$
                $x$.closest := closest_cluster($T$, $x$, $dist(x, w)$)
            else
                $x$.closest := $w$
            relocate($Q, x$)
        }
        else if $dist(x, x.\text{closest}) > dist(x, w)$ {
            $x$.closest := $w$
            relocate($Q, x$)
        }
    }
    insert($Q, w$)
}
end
```

Fig 4. Clustering Algorithm



```
procedure merge($u, v$)
begin
w := $u \cup v$
w.mean := $\frac{|u| u.\text{mean} + |v| v.\text{mean}}{|u|+|v|}$
tmpSet := $\emptyset$
for $i$ := 1 to $c$ do {
    maxDist := 0
    foreach point $p$ in cluster $w$ do {
        if $i = 1$
            minDist := $dist(p, w.\text{mean})$
        else
            minDist := $\min\{dist(p,q) : q \in \text{tmpSet}\}$
        if (minDist $\geq$ maxDist){
            maxDist := minDist
            maxPoint := $p$
        }
    }
    tmpSet := tmpSet $\cup$ {maxPoint}
}
foreach point $p$ in tmpSet do
    $w$.rep := $w$.rep $\cup$ $\{p + \alpha * (w.\text{mean} - p)\}$
return $w$
end
```

Fig 5. Procedure for merging clusters



(a) Data set 1

(b) Data set 2

Fig 6. Date sets

3 Improved CURE:

Our proposed algorithm that reduce the time using Linkage functions and distance measure method that gives good result as compare to CURE clustering algorithm.

Steps of Improved CURE

1) Take numerical data set as an input
2) Use Chebyshev distance for linking weights
3) Construct the tree
4) Now partitioning clusters
5) Apply Clustering using k-Nearest neighbor joining linking.
6) After Partioning merge the partitions.
7) Apply label on outputs.

Linkage Methods:

It identifies the dissimilarity of sets as a function of the pair wise distances of observations in the sets.

1)**Single Linkage:**

The distance between two clusters is smallest distance between an observation in one cluster and other cluster.

$$\mathrm{D}(X,Y) = \min_{x\in X, y\in Y} d(x,y)$$

2) **Average Linkage:**
The distance among two clusters is mean distance between an observation in one cluster and other cluster.

$$D(X,Y) = \frac{1}{|A| \cdot |B|} \sum_{x \epsilon A} \sum_{x \epsilon B} d(x,y)$$

3) **Complete Linkage:**
The distance among two clusters is extreme distance an opinion in one cluster and other cluster.

$$D(X,Y) = \max_{x \in X, y \in Y} d(x,y)$$

4) **Centroid Linkage:**
The distance between two clusters contains one point centre of cluster to other cluster centre points (centroids or means).

4 CURE-NS (New Shrinkage Scheme)

The shrinking scheme of CURE-NS is that the point with low density value shrinks toward the neighboring point with high density, which is based on this fact that the densities around outliers are less than the densities of other areas in the cluster. This assumption is not related to the shape of cluster, so that it is not affected by the shape of cluster as the traditional CURE. CURE-NS is an available hierarchical clustering method that is not dependent on the shape of cluster and less sensitive to outliers.
Instead of using the centroid as the reference of shrinking, for a point being shrunk, we select a point from its neighborhood as reference. The shrinking procedure can be summarized as follows.

1. A fixed number of the scattered points are randomly chosen from the cluster.

2. Calculate the density distribution of the cluster at the positions of the scattered points that is assigning a density value for each scattered point.

3. The initial reference set is defined as empty set.

4. Choose the point with maximal density value as the first point of the reference set. If a scattered point is added to the reference set, this point is removed from the set of scattered points.

5. Compute the distance between the scattered point ***I*** and the reference set R .in which *d,* is the distance between point ***i*** and point *j* that belongs to the reference set.

6. Choose a scattered point **i** with minimal *d,* , and point *j* is the closest point to ***i*** in the reference set. Compare the density values f(i) and f(j) of points **i** and *j* to determine which point should be shrunk.

point I shift toward point ***j*,** if *f (i)* < f(j)
both of point i and **j** do not shift, if f(i) = *f (j)*
point ***j*** shift toward point ***i*,** if f(i) > f(j)

And the distance of shifting is also dependent on their density values.

$$S\ (i,j) = \frac{\alpha\ dij|f(i) - f(j)|}{\max\ (f(i),f(j))}$$

Where **α** is a fraction value.

7. Add the scattered point *i* into the reference set, and re-calculate the position and density value of point ***I*** or *j* due to shrinking.

8. The remaining scattered points are processed as step *5-6* until all points are added into the reference set. The final reference set is used as the set of representative points.

Limitations of CURE

1. CURE ignores the information about the aggregate inter-connectivity of objects in two clusters. So it is introduced Chameleon algorithm.

2. Consider only one point as representative of a cluster.

3. When compared with using all points to describe the cluster, the number of points is reduced, which can decrease the execution times when calculating the distance between two clusters, but the clustering performance with multiple scattered points does not surpass the method with all points.

Limitations of Improved CURE

1. Unfortunately, the assumption that outliers are farther from the centroid of the cluster than normal data points, does not always conforms to reality. Therefore, this shrinking operation results in many problems in dealing with some specific shapes of cluster.

2. Inability to make corrections once the splitting/merging decision is made.

3. Lack of interpretability regarding the cluster descriptors.

4. Vagueness of termination criterion.

5. Prohibitively expensive for high dimensional and massive datasets.

Limitations of CURE-NS.

1. Algorithm can never undo what was done previously.

2. Time complexity of at least O(n^2 *log n*) is required, where *'n'* is the number of data points.

3. Based on the type of distance matrix chosen for merging different algorithms can suffer with one or more of the following:
   i) Sensitivity to noise and outliers.
   ii) Breaking large clusters.
   iii) Difficulty handling different sized clusters and convex shapes.

4. No objective function is directly minimized.

5. Sometimes it is difficult to identify the correct number of clusters by the dendogram.

5 COMPARISION

This table depicts comparison between CURE, Improved CURE and CURE-NS based on different parameters.

Parameters	CURE	Improved CURE	CURE-NS
Time Complexity	O (n^2log n)	O (n^2log n)	O (n^2log n)
Space Complexity	O(n)	O(n^2)	O(n)
Efficiency	Very efficient to identify clusters of non spherical shapes	Very efficient to identify clusters of non spherical shapes	Very efficient to identify clusters of any shapes
Implementation	Complicated	Complicated	Complicated
Sensitivity to outliers	Very sensitive to outliers	sensitive to outliers	Less sensitive to outliers
Does initial partition affects result and runtime?	YES	YES	YES
Optimized For	Separated clusters – Large datasets	Separated clusters – Large datasets	Separated clusters – large High dimensional databases.
Shapes of Clusters	Non-Spherical Shapes i.e. Elongated shapes	Non-Spherical Shapes	Not dependent on the shape of clusters.
Data Structures Used	Heap and K-D Tree	(a,b)-Tree datastructure	Heap and K-D Tree
Distance Measures	Euclidean distance method	Chebyshev distance for linking weights.	Euclidean distance method

6 Conclusions:

In this paper we compared all the three hierarchical clustering algorithms i.e. CURE, Improved CURE and CURE-NS and we found that CURE algorithm whose shrinking scheme is based on the assumption of spherical shape of the cluster, CURE-NS is very computationally efficient for large databases. As opposed to CURE, in CURE-NS the scattered points with low density values shift towards the points with high density values by an evolving reference set. Again we found that CURE-NS completely possesses the ability of identifying arbitrary shapes of cluster and being robust to outliers. In this paper we showed that CURE-NS outperforms the CURE algorithm and also scales well for large databases.

References

1. Sudipto Guha, R. Rastogi, and K. Shim. CURE: A clustering algorithm for large databases. Technical report, Bell Laboratories, Murray Hill, 1997.

2. Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: An efficient data clustering method for very large databases. In Proceedings of the ACM SIGMOD Conference on Management of Data, pages 103-114, Montreal, Canada, June 1996.

3. Komal N. Makadia, Prof. Maulik V. Dhamecha," A SURVEY ON ENHANCING AGGLOMERATIVE HIERARCHICAL TECHNIQUES", International Journal of Advance Engineering and Research Development Volume 1, Issue 11, November -2014.

4. A. Fahad, N. Alshatri, Z. Tariz, A. Alamri, I. Khalil A. Zomaya,, S. Foufou, and A. Bouras, A Survey of Clustering Algorithms for Large Data: Taxonomy & Empirical Analysis‖, IEEE Transactions on Emerging Topics in Computing, Volume: PP 1-12,12 June 2014,ISSN :2168-6750

5. J.Han, M.Kamber. Data Mining: Concepts and Techniques. Academic Press, 2001.

6. G.Karypis, E.H.Han, and V.Kumar, Chameleon: hierarchical clustering using dynamic modeling, Computer, pp.68-77, August 1999.

7. F.M.Frattale, A.Rizzi, M.Panella, G.Martinelli. Scale-based approach to hierarchical fuzzy clustering. Signal processing, 80: 1001-1016, 2000.

8. B .Mirkin, Mathematical Classification and clustering. Kluwer Academic Publishers, Dordrecht, The Netherland, 1996.

9. Marjan Kuchaki Rafsanjani, Zahra Asghari Varzaneh , Nasibeh Emami Chukanlo A survey of hierarchical clustering algorithms. The Journal of Mathematics and Computer Science Vol .5 No.3 (2012) 229-240

10. Ms Komalben N. Makadiya, Prof. Maulik V. Dhamecha **.** An Enhance Approach to Improve CURE Clustering Using Appropriate Linkage Function for Datasets International Journal of Innovative Research in Computer and Communication Engineering. Vol. 3, Issue 6, June 2015.