International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

STCLARanS: An Improved Clustering Large Applications based on Randomized Search Algorithm using Slim-tree Technique

Ricado Q. Camungao

Abstract*: Clustering has been used for data interpretation when dealing with large database in the fields of medicines, business, engineering etc. for the recent years. Its existence paved way on the development of data mining techniques like CLARANS (Clustering Large Applications based on Randomized Search) Algorithm. It is the most efficient k-medoids technique that uses randomized strategy to identify the best medoids in a large dataset. Likewise, it surpasses the clustering performance of both PAM (Partitioning Around Medoids) and CLARA (Clustering Large Applications) in terms of time. This paper addresses the task of integrating Slim-tree method to CLARANS for the development of the proposed Slim-tree Clustering Large Applications based on Randomized Search (STCLARanS) Algorithm and an experimental evaluation was prepared using synthetic and real datasets for the comparison of the quality of the clustered output of the CLARANS and the proposed STCLARanS algorithms. The Slim-tree method is used for pre-clustering of the objects in the dataset in identifying the objects in the middle level as the sample objects used to start the clustering process. The proposed Algorithm assumes that with the new sampling strategy to draw the initial cluster centers to start the clustering process may yield to better quality of the clustered outputs as compared to the clustered output of the CLARANS algorithm. The quality of the clustered output is measured on the accumulated distances of the objects to their cluster centers.

Index Terms: Clustering, k-medoids, CLARANS, Slim-tree, randomized search, pre-clustering. STCLARanS.

I. INTRODUCTION

Clustering is a method of grouping objects or data into number of groups of related objects according to their numerical value. Primarily, clustering divides data into group of objects that are alike and dissimilar, where objects within the cluster are similar but different to the objects of the other cluster [1]. Among the clustering Algorithms developed for large spectrum of applications, Partitioning Algorithm is the simplest and most popular that includes k-means and k-medoids Algorithms. The k-medoids based Algorithms are more robust, since they are less sensitive to the existence of outliers, can be applied to large database, do not depend on the input order of the dataset and invariant to change and orthogonal transformation of objects [2] but very time - consuming when applied to large dataset.

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author
Ricardo Q. Camungao, Associate Professor III and designated as the Dean of the College of Computing Studies, Philippines.

The best k-medoids algorithms that deals with large databases is the Clustering Large Application based on Randomized Search (CLARANS) Algorithm. In identifying the best representative, the CLARANS use randomized strategies to select the initial cluster center to start the clustering process.
The clustering performance of the CLARANS algorithm in terms of time as compared to the two other K-medoids algorithm the PAM (Partitioning Around Medoids) and CLARA (Clustering Large Applications), the former outperform both PAM and CLARA. Nevertheless, quality of the clustered output is still one of the drawbacks of the CLARANS algorithm because of the randomized search strategy. It is in this light, that this paper attempts to improve the clustering output of the CLARANS algorithm through integrating Slim-tree method in its clustering process. The Slim-tree method is used for the sampling technique in the identification of the initial cluster centers. For the realization of this undertaking, the following are need to sought for; (1) the integration of the Slim-tree technique in the CLARANS Algorithm for the development of the improved CLARANS algorithm named as STCLARanS (Slim – Tree Clustering Large Application using Randomized Search) Algorithm and (2) experimental evaluation using synthetic and real-world datasets to test the clustering output of the proposed algorithm in terms of the accumulated distances of the objects to their cluster centers.

II. RELATED LITERATURE

A. Clustering

Clustering techniques was introduced in the past decades and its primary goal is to identify structures or clusters present in the data [8]. By definition, clustering is a branch of Statistics that has been intensely studied and applied to many applications [9],[10],[11] it primarily aims to identify and extract significant groups in underlying data [12]. Among the clustering Algorithms developed for a large spectrum of applications of the partitioning algorithms [13] is the simplest and most popular one which includes k-means and k-medoids. The k-medoids [14] primarily aim to find a none-overlapping set of clusters, where each cluster has one representative object or cluster center that is most centrally located in the cluster considering a dissimilarity or distance measure (Euclidean distance) hence, finding the shortest distance between the objects to the cluster centers is necessary [13]. In addition, Euclidean distance calculate the distance between objects to cluster centers [14] as in eq. 1 and a common measure of goodness is the sum of squares of the direct Euclidean distance between the object and their cluster centers [1] using eq. 2.

$$d(x,y) = \sqrt{(x1-y1)^2 + (x2-y2)^2} \qquad (1)$$

Where: *x* and *y* are the objects, *x1* and *x2* are coordinates of object *x*, and *y1* and *y2* are coordinates of object *y*. The k-medoids Algorithms have two main phases these are;

1. Initialization – initial set of number (k) objects are selected as medoids.
2. Evaluation – It tries to minimize an object function usually based on the sum of the total distance among non – selected objects and their medoids, i.e. the evaluation step tries to minimize:

$$\sum_{j=1}^{n} d(ri, sj) \qquad (2)$$

B. CLARANS (*Clustering Large Applications using Randomized Search*) Algorithm

The development of CLARANS was based in the context of spatial data mining. The randomized search strategy of the CLARANS Algorithm greatly improved its efficiency (computational complexity or time) and effectiveness (average distortion over the distances). The process of searching for the best cluster centers or medoids during the evaluation step, CLARANS randomly search the representative objects from the remaining number of objects (n-k) in the dataset. The parameter provided by the user (*maxNeighbor*) is used to control the number of representative objects tried in clustering the objects. After the number of attempts set (*maxNeighbor*), and no better solution is found the local optimal is assumed to be reached. The process continues until *numLocal* local *optimals* have been found [1]. According to [3] it was suggested that the parameters *numLocal* be set to 2 and *maxNeighbor* (250, 1.25% of k(n-k)). The values of these parameters were obtained from the experimental results of determining the number of *numLocal* and *maxNeighbor* [3]. The result shows that setting the value of *numLocal* greater than two is not cost – effective for the reason that the increase in quality is insignificant. This is an indication that setting the typical local minimum to 2 can provide very high quality of clustering output. The value of the *maxNeighbor* is based on the formula: if $k(n-k) \leq minmaxneighbor$ then $maxNeighbor = k(n-k)$; otherwise, *maxNeighbor* equals the larger value between *p% of k(n-k)* and *minmaxneighbor*. The *minmaxneighbor* is the threshold equal to 250 and to keep a good balance between runtime and quality the value of *p* is 1.25%.

C. Slim-tree

The basic structure of many metric trees, such as M-tree [5], Slim-tree [6] and DBM tree [5], divides the data space into regions using representatives to which the other objects in each group is associated with. The Slim-tree is a balanced and dynamic tree that grows bottom-up from the leaves to the root. Same with other metric trees, each object in the dataset is grouped into fixed size that corresponds to a tree node [6] where objects are stored in the leaves. Its primary intent is to arrange the objects in a hierarchical structure using a representative as the center of each minimum bounding region which covers the objects in a subtree. The Slim-tree has two kinds of nodes; data or leaves nodes and index nodes. The size of a page is fixed; each type of node holds a predefined maximum number of objects. It assumes that the capacity of the leaves is equal to the number of the index nodes.

The structures of data or leave and index nodes are as follows [9];

leafnode [array of $<Oid_i, d(S_i,S_{rep}),S_i>$]

Where Oid_i, is the identifier of the object S_i $d(S_i,S_{rep})$ is the distance between the object S_i and the representative object of this leaf node S_{rep}.[9]

indexnode [array of $<S_i, R_i, d(S_i,S_{rep}), Ptr(TS_i), NEntries(Ptr(TS_i))>$]

Where S_i keeps the object that is the representative of the subtree pointed by $Ptr(TS_i)$ and R_i *is* the covering radius of the region. The distance between S_i and the representative of this node S_{rep} stored in $d(S_i,S_{rep})$. The pointer by $Ptr(TS_i)$points to the root node of the subtree rooted by S_i. The number of entries in the node pointed to by $Ptr(TS_i)$ is held by $NEntries(Ptr(TS_i)$[9]. Figure 1 and 2 shows the Slim-tree structure.

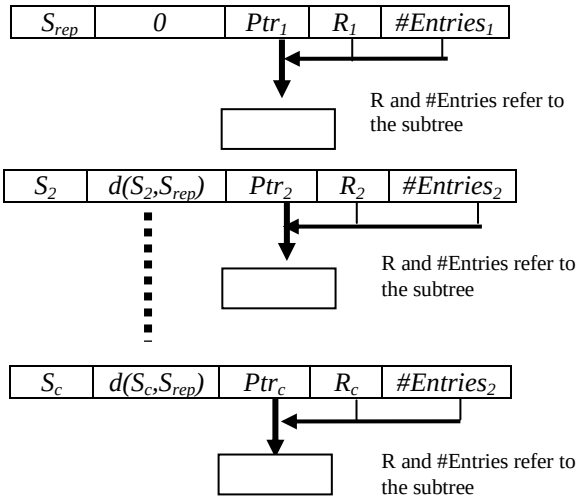

Fig. 1: Representation of the memory structure on index node of the Slim-tree

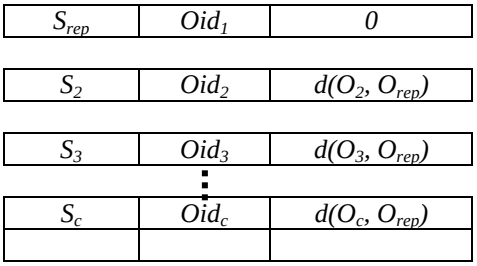

Fig. 2: Representation of the memory structure on leaf node of the Slim-tree

To build a Slim-tree, it is essential to make use of the *ChooseSubtree* Algorithm as strategy in inserting new object when more than one node is eligible for adding new object. The Slim-tree has three options for *ChooseSubtree* Algorithm [5],

1. *Random:* this randomly chooses one of the qualifying nodes.
2. *MinDist:* it chooses the node that has the minimum distance from the new object and the center of the node.
3. *MinOccup*: it chooses the node that has the minimum occupancy among the qualifying nodes.

The strategy plays an important role for the reason that it influences the appearance of the constructed tree as regards to the amount of overlapping of nodes. According to [2] the *minOccup* option produce high rate of node occupation that yields to fewer number of disk accesses, however the degree of overlapping of nodes is high; whereas, the *minDis*t option produce lower rate of node occupation and produce taller trees with a lower degree of overlapping of nodes [6].

The generated Slim-tree in a sense is a number of clusters comprising of objects and the node capacity. However, the number of clusters is not a parameter given by the user to generate the tree so it cannot be considered as a clustered output. Nevertheless, each representative objects in the index nodes can be considered as approximate cluster centers on each level of the tree. Therefore, it is possible to use these objects as initial cluster centers to group the data [2].

The default Algorithms in building the Slim-tree are as follows; *"minoccup"* option for the *ChooseSubtree* Algorithm, minimum spanning tree (MST) strategy for splits and the capacity node *C* is 60 for vector (L_2 distance) [7].

III. THE SLIM – TREE CLUSTERING LARGE APPLICATIONS BASED ON RANDOMIZED SEARCH (STCLARANS) ALGORITHM

The computational complexity of the CLARANS Algorithm presented in the previous section is on randomized search of cluster centers used in clustering objects in the dataset. The randomization of the initial cluster centers in a large database provide probability of not choosing the best initial representatives to start the clustering process which may result to a low quality of clustered output.

In order to determine the best cluster centers that provide a better quality of clustered output, a frequent approach is to identify sample objects from the original database as the starting objects for the initial clustering process. This made possible through employing the concept of the Slim-tree method in the CLARANS Algorithm, to identify the sample objects use in the identification of the initial cluster centers.

The strategy is to determine the sample objects from the output of the constructed tree. The assigned cluster center is the center object of a particular sub tree and each level of the constructed tree represents a data space division containing all the objects in the dataset. However, not all representative nodes of the constructed tree can be considered as sample objects, a certain part of the tree is viable to consider as sample object. The foregoing scenario, leads to an assessment on which part of the tree will best represent the objects as to data distribution is concern and as to which part can generate better quality of clustering output [2].

According to [2] the best preference to consider the objects to use for sample objects are the middle level of the constructed tree because the distribution of the objects encloses enough information that can result to a better quality of clustering. Whereas, objects in the top and bottom part of the constructed tree are not viable to use as sample objects for the reason that the top level contains less number objects are found, hence less information about the data distribution and the lower level contains more objects than the top and middle level of the tree. Considering the objects in this level may result to a poor performance of the Algorithm.

The output of the Slim-tree is used as sampling strategy and it is integrated into the CLARANS Algorithm and named as STCLARanS Algorithm.

The proposed STCLARanS Algorithm is depicted as follows;

Initialization phase

1. Input number of cluster (k) [1].
2. Initialize mincost to a large number, numlocal to 2 and determine the maxneighbor is (250, 1.25% of k(n-k)) [1].

Clustering phase

Slim-tree computation

3. Input parameters for building the tree: the page size and ChooseSubTree Algorithm.
4. Build the Slim-tree.
5. Consider the middle level of the tree as the **sample objects (D)**.

6. Set *current* to an arbitrary node from the **sample objects (D)**.
7. Set j =1;
8. Consider a random neighbour (S) of *current*, calculate the cost coefficient of the two nodes.
9. If S has a lower cost, set current is equal to S, and proceed to step 7.
10. If not, when j is greater than maxneighbor, compare the cost of *current* with *MinCost*. If the *current* is less than the *mincost* then, *MinCost* is equal to the cost of *current*, and *bestnode* is equal to *current*.
11. Add one (1) to the current value of i. If i > numlocal, output bestnode and halt. If not, proceed to step 6.

The STCLARanS Algorithm has two main phases, these are the initialization and clustering phase.

The **initialization phase** enables the assigning and inputting of the initial values of the needed parameters for clustering such as number of cluster (k), *numLocal* to 2 and identifying the values of the *MinCost* and *maxNeighbor (*250, 1.25% of k(n-k)) [1].

The **clustering phase** generates the clustered output of the given set of data. The initial step is to build the Slim-tree. The middle level of the constructed tree is used as the sample objects (D). The final step of this phase is the clustering process, the initial representatives or cluster centers are arbitrarily chosen from the sample objects (*D).* The number of objects tried in this phase is dependent on the number or value of *maxneighbor* set in the initialization phase. The value of *maxneighbor* corresponds on the number of attempts in identifying the best sets by way of accumulating the distances (*MinCost*) of the objects to their cluster centers. After the number of attempts and the value of the *numlocal* is greater than 2, the best sets are identified having the lowest *MinCost*. To end with, cluster the objects of the dataset using cluster centers of the best sets.

IV. EXPERIMENTAL EVALUATION

To determine the quality of the clustered output using equation (1) of the CLARANS and the proposed STCLARanS algorithms in terms of the closeness of the objects to their cluster centers. An experimental evaluation was made to simulate their performance using synthetic and real datasets as presented in Table I. The CLARANS and proposed STCLARanS Algorithms were implemented in the same platform using the C# language and to obtain a fair comparison the simulations were performed in a PC with an Intel core i5 2.7 Ghz CPU, 4 GB RAM and 500GB hard disk with 200 GB free disk space. The clustering output of the Slim-Tree Clustering Large Applications using Randomized Search (STCLARanS) Algorithm was shown in figure 4 whereas, sample clustering output of Clustering Large Applications using Randomized Search (CLARANS) algorithm was presented in figure 5. The sample clustering output presents the phases of the two algorithms, this made possible by converting the phases to its equivalent C# program. Also, to compare the quality of the clustered output of each Algorithm a variable *MinCost* was used to accumulate the total distances of the objects to their cluster centers and a time function was also included in the program to measure the time spent in clustering the objects. The *MinCost* and time was recorded and displayed at the bottom part of fig. 3 and fig. 4.

```
SLIM TREE RESULT: 869 Object/s

The best sets are:

89.1,82.3
80.7,63.4


CLARANS RESULT


CLUSTERS                CLUSTER CENTERS                 NUMBER OF OBJECTS
Cluster #: 1            89.1,82.3                                5501
Cluster #: 2            80.7,63.4                                3557


MinCost: 123554.41  Execution Time: 00:00:17.9495100
```

Fig. 3: Sample clustering output of the proposed Slim-Tree Clustering Large Applications based on Randomized Search (STCLARanS) Algorithm

```
The best sets are:

84.6,75.6
99.5,71.2


CLARANS RESULT


CLUSTERS                CLUSTER CENTERS                 NUMBER OF OBJECTS
Cluster #: 1            84.6,75.6                                7783
Cluster #: 2            99.5,71.2                                1275


MinCost: 141670.56  Execution Time: 00:02:25.8488849
```

Fig. 4: Sample clustering output of the Clustering Large Application based on Randomized Search (CLARANS) Algorithm

Three datasets used in the experiment; the first two were composed of synthetic data derived by the authors composed of x and y coordinates ranging from 50.00 to 99.99 and the third dataset was a real data taken from the result of the faculty performance evaluation by the student in two semesters. These datasets were used to enable a thoughtful assessment on the quality of the clustered output of the Algorithms in clustering objects in the dataset based on the number of *k* clusters as shown in Table I.

Table I. Description of Synthetic and Real Datasets

Name	Total number of Objects	No. of *k* clusters	*d()*
Dataset1_6k	6,000	5	L_2
Dataset2_9k	9,058	5	L_2
Faculty Performance Evaluation Rating (FPER)	651	5	L_2

A. CLARANS versus the proposed STCLARanS Algorithms

The experiments were carried out based on the quality of the clustered output using equation (1) of the CLARANS and proposed STCLARanS Algorithms in clustering objects in the datasets. Both Algorithms used random samples; the difference is that the proposed Algorithm draws initial nodes in the middle level of the constructed tree that compose of actual cluster centers whereas CLARANS arbitrarily select initial nodes from the given dataset which has a greater number as compared to the number of objects in the middle level of the constructed tree. The number of executions of the Algorithms (CLARANS and STCLARanS) is based on the values of numlocal and *maxneighbor* discussed in Section 3 and *MinCost* is computed using Equation 2 as presented in section II. The time format to measure the clustering performance is hours: minutes: seconds (00:00:00). The comparison of the experiments shows in Tables 2 and 3. That, in Table 3 the proposed STCLARanS Algorithm spent less time than the CLARANS Algorithm in clustering objects of the three datasets and in Table 2 presents that the STCLARanS accumulated lower values of *MinCost* as compared to CLARANS Algorithm. The value of the *MinCost* provides evidence on the closeness of the clustered objects to the cluster centers which implies that the lower the value of the *MinCost* acquired the better quality of the clustered output [1].

Table II. CLARANS and STCLARanS quality of clustered objects (*mincost*)

Datasets	No. of Objects	CLARANS Algorithm (*MinCost*)	STCLARanS Algorithm (*MinCost*)
Real Dataset	651	1279.94	1217.90
Synthetic_6k	6000	78100.14	72877.71
Synthetic_9k	9058	141670.56	123554.41

Table III. CLARANS and STCLARanS clustering performance in terms of time

Datasets	No. of Objects	CLARANS Algorithm (Time)	STCLARanS Algorithm (Time)
Real Dataset	651	00:00:09.004	00:00:01.376
Synthetic_6k	6000	00:01:39.001	00:00:51.069
Synthetic_9k	9058	00:02:25.040	00:00:17.950

Since, the sampling strategy considers the middle level of the constructed tree which in a real sense is cluster centers. The STCLARanS Algorithm draws initial representative objects from the sample objects where each object is convenient to use to start the clustering process. As a result the STCLARanS Algorithm accumulates lower computational cost or *MinCost* using the synthetic and real dataset presented in Table 2.

V. CONCLUSIONS AND FUTURE WORKS

This paper presented an improved clustering large applications using randomized search Algorithm named as STCLARanS Algorithm. The STCLARanS Algorithm employs Slim-tree method to pre – cluster objects in the dataset. The integration of the Slim-tree technique as basis for the sampling strategy is effective in identifying the initial cluster centers for clustering the objects, this is manifested on the accumulated distances or *MinCost* of the objects to their cluster centers as presented in table 2 and 3 where the STCLARanS Algorithm produce lower value of the *MinCost* which implies that the lower the accumulated distances of the objects to their centers the better quality of clustered objects. Furthermore, the proposed algorithms spent lesser period of time in clustering a given dataset than the CLARANS algorithm. With the conclusions derived based on the findings, it is further recommended to develop a windows – based simulator of the proposed STCLARanS Algorithm and use as alternative tool for data interpretation. Also, for further enhancements of the proposed Algorithm, Apply bloat – factor in tightening the resulting Slim-tree. The bloat – factor resolve the issues on overlapping of objects in the constructed tree that measure the goodness [4], [5] of the constructed tree..

REFERENCES

1. Ng, R.T., and Han, J. (2002). Clarans: A method for clustering objects for spatial data mining. IEEE Transactionson Knowledge and Data Engineering (TKDE), 14(5):1003-1016
2. Barioni, M.C., Razente, H., Traina, A,J, and Traina, C. (2001). An Efficient Approach to Scale up k-medoid based Algorithms in Large Databases. XXI Simposio Brasileiro de Banco de Dados, pages 265 – 278.
3. Vieira, M. R., Chino, C. T. F., and Traina, A. J. M. (2004). DBM tree: A metric access method sensitive to local density data. In Brazilian Symposium on Databases (SBBD), pages 163 – 177.
4. Traina, A.J. M., Jr., C.T., Bueno, J.M., and de A. Marques, P.M. (2002). The metric histogram: A new and efficient approach for content –based image retrieval. In IFIP working Conference on Visual Database System (VDB), pages 297-311.
5. Traina Jr., C., Traina, A., Faloustsos, C., and Seeger, B. (2002).Fast Indexing and Visualization of Metric Data Sets Using Slim-Trees.Vol 14. No.2 (244 – 260)
6. Han, J. and Kamber, M. (2001). Data Mining: Concepts and Techniques. Academic Press, San Diego, CA.
7. Kasuga, H., Yamamoto, H. and Okamoto, M., Color Quantization using the Fast K – Means Algorithm, Systems and Computers in Japan, vol. 31, no. 8, pp. 33 – 40, 2000
8. He, Z. and Xiong, F., A Constructed Partition Model and K – Means Algorithm, Journal of Software, vol. 16, no. 5, pp. 799 – 809, 2005
9. Weiwei, N., Jieping, L. and Zhihui, S., An Effective Distributed K – Means Clustering Algorithms based on the Pretreatment of Vectors Inner Product, Journal of Computer Research and Development, vol. 42, no.9, pp. 1493 – 1497, 2005
10. Fattouh, L., Karam, O., El Sharkawy, M and Khaled, W., Clustering for Network Planning, WSEAS Transactions on Computers, Issue 1, Volume 2, ISSN 1109 – 750, January 2003.
11. Bradley, P., Fayyad, U. and Reina, C. (1998). Scaling clustering algorithms to large databases, Proceedings 4th KDD, pp. 9 – 15.
12. Kaufman, L. and Rousseuw, P.J. (2005), Finding groups in data: An introduction to cluster analysis. John Wiley and Sons.
13. Ahmad, T., A Shortest Path Method on a Surface in Space. International Journal of Advance Trends in Computer Science and Engineering, Article 16 of Vol. 8 No. 2 ,March – April 2019.
14. Chandirika, B., Sakthivel, N.K., Subasree, S., An Energy Efficient K-Means Clustering Based Trust Model for Wireless Sensor Networks. International Journal of Advance Trends in Computer Science and Engineering, Article 8 of Vol. 8 No. 2 ,March – April 2019.

AUTHORS PROFILE

Ricardo Q. Camungao, currently holds a rank of Associate Professor III and designated as the Dean of the College of Computing Studies, Information and Communication Technology of Isabela State University, finished his Doctor in Information Technology at the Technological Institute of the Philippines on April 10, 2017, he has been tapped by various agencies to perform Job relevant to his field of specialization. He is a member of the Philippine Society of IT Educator, Philippine Computer Society and PSSUCES. His field of interest includes information system and data mining.