Optimization of Clustering Algorithm Using Metaheuristic

Ayushi Sinha[1], Mr. Manish Mahajan[2]

[1]M.Tech Student, Graphic Era University, Dehradun, India

[2]Associate professor, Computer Science & Engineering Department, Graphic Era University, Dehradun, India

ayushi.sinha007@gmail.com, *Manish.mahajan@outlook.com*

Abstract A vital issue in information grouping and present a few answers for it. We explore utilizing separation measures other than Euclidean sort for enhancing the execution of Clustering. We additionally build up another point symmetry-based separation measure and demonstrate its proficiency. We build up a novel successful k-Mean calculation which enhances the execution of the k-mean calculation. We build up a dynamic linkage grouping calculation utilizing kd-tree and we demonstrate its superior. The Automatic Clustering Differential Evolution (ACDE) is particular to Clustering basic information sets and finding the ideal number of groups consequently. We enhance ACDE for arranging more mind boggling information sets utilizing kd-tree. The proposed calculations don't have a most pessimistic scenario bound on running time that exists in numerous comparable calculations in the writing.

Experimental results appeared in this proposition exhibit the viability of the proposed calculations. We contrast the proposed calculations and other ACO calculations. We display the proposed calculations and their execution results in point of interest alongside promising streets of future examination.

Keywords: *clustering, Data mining, k-mean, ACO*

*****

I. INTRODUCTION

A metaheuristic (Meta: in an upper level, Heuristic: to discover) [1] is formally characterized as an iterative era process which controls a subordinate heuristic by consolidating keenly diverse ideas for investigating and exploiting the search space. Learning procedures are utilized to structure data with a specific end goal to discover productively close ideal arrangements. Metaheuristic calculations [2] are surmised and more often than not nondeterministic methods which constitute metaheuristic calculations extending from straightforward neighborhood look strategies to complex learning forms.

Clustering [3] is a division of information into gatherings of comparable items. Every gathering, called group comprises of articles that are comparative inside the cluster and unlike objects of different groups. Speaking to information by fewer groups essentially loses certain fine points of interest, yet accomplishes rearrangements, and thus might be considered as a type of information pressure. It speaks to numerous information objects by few groups' models information by its Clusters. Information's are displaying places grouping in a recorded point of view which is established in science, insights, and numerical examination. Grouping is the subject of dynamic exploration in a few fields, for example, measurements, design acknowledgment, manmade brainpower, and machine learning. From a handy point of view, Clustering assumes an extraordinary part in information mining applications, for example, logical information investigation, data recovery and content mining, spatial database applications, Web examination, showcasing, medicinal diagnostics, computational science, and numerous others.

The Clustering issue has been tended to in numerous settings and by specialists in numerous orders. This mirrors its wide request and value as one of the progressions in exploratory information examination. Despite the fact that order [4] is a successful means for recognizing gatherings or classes of items, it requires the regularly unreasonable accumulation and marking of an extensive arrangement of preparing tuples or examples, which the classifier uses to demonstrate every gathering. It is frequently more attractive to continue in the opposite heading: First parcel the arrangement of information into gatherings in view of information closeness (e.g., utilizing Clustering), and afterward dole out marks to the generally little number of gatherings. From a machine learning point of view Clusters relate to concealed examples, the quest for groups is unsupervised learning [5], and the subsequent framework speaks to an information idea. In this manner, Clustering is unsupervised learning of a concealed information idea. Information mining manages extensive databases that force on Clustering investigation extra extreme computational prerequisites. These difficulties prompted the development of effective comprehensively pertinent information mining Clustering techniques.

II. CLUSTERING ALGORITHMS

There are a huge number of grouping strategies one can experience in the writing. The majority of the current information grouping calculations can be named various leveled or partitional as appeared in Figure 1.1. Inside every class, there exists an abundance of sub-class which incorporates diverse calculations for finding the groups.

While various leveled calculations [13] fabricate Clusters bit by bit (as precious stones are developed), dividing calculations [14] learn groups straightforwardly. In doing as such, they either attempt to find Clusters by iteratively moving focuses between subsets, or attempt to recognize groups as ranges exceptionally populated with information.

Thickness based calculations [15] ordinarily view groups as thick areas of articles in the information space that are isolated by districts of low thickness. The primary thought of thickness based methodology is to discover districts of high thickness and low thickness, with high-thickness areas being isolated from low-thickness locales. These methodologies can make it simple to find subjective groups.

As of late, various Cluster calculations have been introduced for spatial information, known as lattice based calculations. They perform space division and after that total suitable sections [16].

Numerous other Cluster systems are created, essentially in machine realizing, that either have hypothetical criticalness, are utilized customarily outside the information mining group, or don't fit in already delineated classifications.

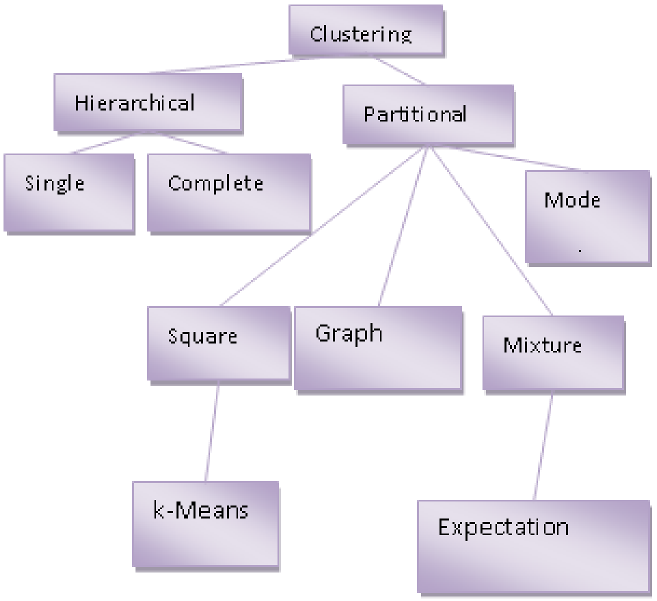

Figure 1.1: Taxonomy of clustering approaches (adapted from Jain, Murty, and Flynn [24])

III. METHODOLOGY K-MEANS ALGORITHM

K-Mean utilizes a two-stage iterative calculation to minimize the aggregate of point-to centroid separations, summed over all k groups: The principal stage is "cluster" upgrades, where cycle comprises of reassigning focuses to their closest group centroid, at the same time, trailed by recalculation of cluster centroids. The second stage utilizes "online" overhauls, where focuses are independently reassigned. By doing as such will lessen the total of separations, and cluster centroids are recomputed after every reassignment. The emphasis amid this second stage comprises of one go through every one of the focuses. K-means can merge to a neighbourhood ideal which is a parcel of focuses in which moving any single point to an alternate cluster expands the aggregate whole of separations [23].

The K-Mean Algorithm is exhibited as takes after:

- Initialize K focus areas (C1, ..., Ck).
- Assign every information guide Xi toward its closest group focus Cj.
- Update every group focus Cj to be the mean of all Xi that have been doled out as nearest to it.
- Calculate summation of least separation measures
- If the estimation of D has merged, then return (C1, ..., CK); else go to Step 2.

In this manner k-Mean has a hard participation capacity. Besides, k-Mean has a steady weight capacity, i.e. all examples having a place with a group have risen to impact in registering the centroid of the cluster. The k-Mean has two fundamental focal points [24]:

- It is anything but difficult to actualize.
- The time multifaceted nature is just O (n) (n being the quantity of information focuses), which makes it appropriate for huge information sets.

However the k-Mean experiences the accompanying drawbacks:

- The client needs to indicate the quantity of classes ahead of time.
- The execution of the calculation is information subordinate.
- The calculation utilizes an insatiable approach and is vigorously subject to the underlying conditions. This frequently drives k-intends to unite to imperfect arrangements.

Stephen J. Redmond and Conor Heneghan [25] displayed a strategy for instating the K-Mean clustering calculation utilizing kd-tree. The proposed strategy relies on upon the utilization of a kd-tree to play out a thickness estimation of the information at different areas. They utilized an alteration of Katsavounidis' calculation, which joins this thickness data, to pick K seeds for the K-Mean calculation.

K. Mumtaz1 and K. Duraiswamy [26] proposed a novel thickness based k-Mean clustering calculation to beat the disadvantages of DBSCAN and k-Mean grouping calculations. The outcome is an enhanced adaptation of k-means grouping calculation. This calculation performs superior to anything DBSCAN while taking care of groups of circularly dispersed information focuses and marginally covered clusters. Yet, there is a confinement for this calculation. It requires an earlier particular of a few parameters, and the clustering execution is influenced by these parameters.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is an information clustering calculation proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996 [27]. It is a thickness based clustering calculation since it finds various groups beginning from the assessed thickness appropriation of comparing hubs. DBSCAN is a standout amongst the most widely recognized clustering calculations furthermore most referred to in exploratory writing.

i. Pseudo-Code of the ACDE Algorithm

The pseudo code of the complete ACDE algorithm is presented The pseudo code of the complete ACDE algorithm is presented below.

Step 1: Initialize each search variable vector in DE to contain k number of randomly selected cluster centers and k (randomly chosen) activation thresholds in [0, 1].

Step 2: Find out the active cluster centers in each chromosome with the help of the rule described.

Step 3: For iter =1to MAXITER do/MAXITER is most extreme number of emphasess

Check if the quantity of information focuses having a place with any group is under 2. Provided that this is true, redesign the cluster focuses of the chromosome utilizing the idea of normal depicted before.

Change the populace individuals as indicated by the DE calculation with adjustments proposed in segment 2.3.2.4. Utilize the wellness of the vectors to control the development of the populace.

Step 4: Report as the last arrangement the cluster focuses and the segment got by the all around best vector (one yielding the most noteworthy estimation of the wellness capacity) at iter = MAXITER

This segment depicts our commitment for enhancing proficiency of k-means. We called the proposed calculation a novel powerful k-Mean calculation. We utilized an enhanced PS-Based separation measure for building up the proposed calculation. We exhibit the pseudo code of a novel Effective K-Mean calculation that we have created as follows:3

```
1. Initialize K center locations (C1, ..., CK).

2. Select DPs of kd-tree.

3. FOR each cluster center Cj do

      FOR each data point Xi do

            Calculate dIPS (Xi , Cj) by using Equation 3.1.

     END FOR

END FOR

4. Assign each data point Xi to its cluster center Cj by selecting the minimum distance of dIPS (Xi , Cj).

5. Update each cluster center Cj as the mean of all Xi that have been assigned to it.

6. Calculate

7. If the value of D has converged, then return (C1, ..., CK); else go to Step 3.
```

This algorithm has three main advantages:

1) It is very easy to implement.

2) It doesn't utilize extra parameters like different calculations which are proposed in writing for enhancing the proficiency of K-means calculation. The greater part of parameters which are utilized by different calculations are touchy to the execution of grouping.

3) Its execution is superior to the execution of K-means. It characterized more information sets which were arranged erroneously by K-Mean calculation. In any case this calculation experiences the accompanying burdens:

1) The user has to specify the number of classes in advance.

2) Processing time is increased compared to k-means using Euclidean distance.

3) The algorithm uses a greedy approach and is heavily dependent on the initial conditions. This often leads the results to converge to sub-optimal solutions.

We propose to use in step 1 of our novel algorithm to eliminate the dependency on the initial conditions.

ii. ACO Based Cluster Refinement

Ant-based clustering and sorting was initially presented for assignments in apply autonomy. The changed the calculation to be material to numerical information investigation, and it has in this manner been utilized for information mining, diagram parceling and content mining. Such Ant-based insect based techniques have demonstrated their viability and effectiveness in some experiments. Be that as it may, the Ant-based insect based grouping methodology is by and large juvenile and leaves huge space for upgrades. With these contemplations, be that as it may, the standard Ant-based clustering performs well; the calculation comprises of parcel of parameters like pheromone, specialist memory, number of operators, number of emphasess and group recovery and so on. For these parameters more suppositions have been made in the past works. In this way, ants are utilized to group the information focuses. Here, surprisingly we have utilized ants to refine the clusteres. The clusteres from the above segment are considered as contribution to this ACO based refinement step.

The fundamental explanation behind our refinement is, in any clustering calculation the got groups will never give us 100% quality. There will be a few blunders known as mis clustering. That is, an information thing can be wrongly grouped. These sorts of blunders can be stayed away from by utilizing our refinement calculation.

This Ant-based is permitted to go for an arbitrary stroll on the groups. At whatever point it crosses a cluster, it will pick a thing from the group and drop it into another group while moving.

IV. RESULT SIMULATION

For the comparative study of refinement of clusters from k-means with ant colony optimization, i have taken total number of cities as twenty and total number of iterations as two hundred fifty. The simulated work is performed on Matlab on Intel core i3 processor.

- **ANT Colony Optimization Result**

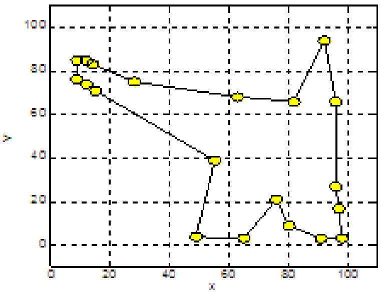

Figure 1.2: Result of optimization using ACO

In the above figure the yellow dots shows the location of coordinates of twenty cities. The way all the twenty cities are connected to each other shows the optimized path.

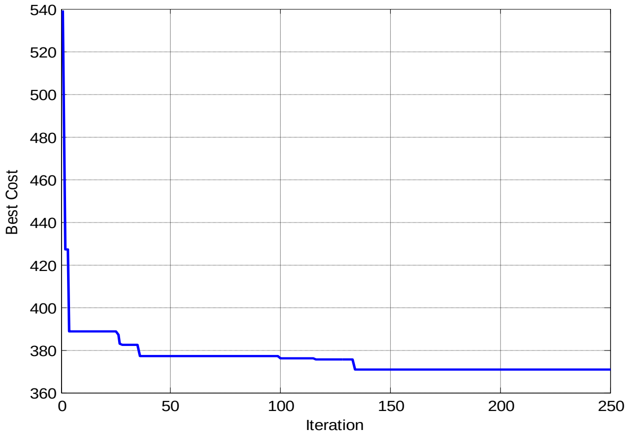

Figure 1.3: Best cost (minimum distance) vs number of iteration

The above graph is plotted between best costs versus number of iterations. In the above graph it is clearly visible that the best cost is coming 370.7421 in only 134 iterations.

- **K-Mean Cluster**

For K-mean cluster we used a cluster toolbox inside the Matlab.

Clustering Toolbox used in Matlab.

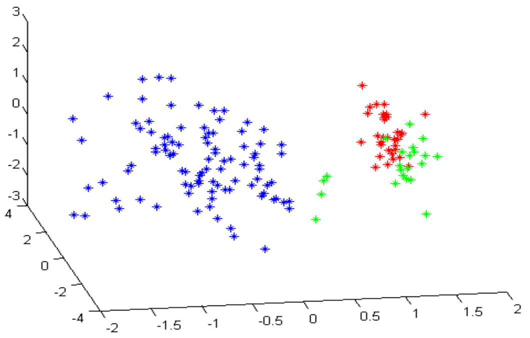

Figure 1.4: Result of optimization using K-mean cluster

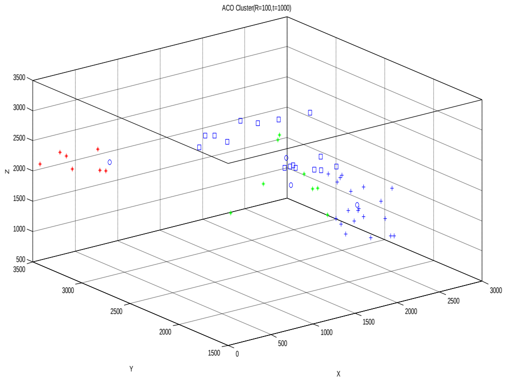

Figure 1.4: Result of optimization using K-mean cluster with ACO (Proposed Algorithm).

TABLE 1. Simulation metrics

	Proposed	Hierarchical	K-Mean
Total no. of iterations	250	250	250
Total no. cities	20	20	20
Max. Cost	539.3059	673.4477	960.5434
Best cost	370.7421	370.7421	457.3522
Best cost in no. of iterations	134	148	206

VI. CONCLUSION

We have proposed Ant Colony Optimization (ACO) calculation to enhance the cluster quality from k-Mean calculation. At to start with, the underlying cluster focuses are chosen in light of measurable mode based count to merge to a superior neighbourhood least. Furthermore, in the second step, we have proposed a novel strategy to enhance to cluster quality by subterranean insect based refinement calculation. The proposed calculation is tried in therapeutic space and the trial results demonstrate that refined beginning stages and post preparing refinement of clusters gives preferred results over the customary calculation.

In this paper we depicted a crucial issue in information grouping and exhibited a few answers for it. We researched utilizing separation measures other than Euclidean sort for enhancing the execution of grouping. We additionally built up another separation measure and demonstrated its productivity. We built up a novel powerful k-Mean calculation which enhanced the execution of the k-mean calculation. We built up a novel clustering calculation by utilizing kd-tree and we demonstrated its execution. The ACO calculation that we introduced is particular to clustering basic information sets and finding the ideal number of groups consequently. We enhanced ACO for grouping more perplexing information sets utilizing kd-tree. The proposed calculations did not have a most pessimistic scenario bound on running time. In the above experimental results we conclude that K-Mean with ant colony optimization produces the best solution of 370.7421 in just 134 iterations out of 250 iterations taken. The K-mean produces the same result in 148 iterations while the genetic algorithm produces very high cost of 457.3522 in 206 iteration. Therefore ant colony optimization is the best algorithm in respect of best solution versus number of iteration.

REFRENCES

[1] Osman, and G. Laporte, “Metaheuristics: A bibliography,” Annals of Operations Research Journal, Springer Netherlands, vol. 63, no. 5 , pp. 513–623, October 1996.

[2] C. Blum, and A.Roli, “Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison,” ACM Computing Surveys, vol. 35, Number 3, pp. 268–308, September 2003.

[3] G. Gan, C. Ma, and J. Wu, Data Clustering: Theory, Algorithms, and Applications. ASA-SIAM Series on Statistics and Applied Probability, SIAM, Philadelphia, ASA, Alexandria, VA, 2007.

[4] S. K. Halgamuge, and L. Wang, Classification and Clustering for Knowledge Discover. Springer-Verlag Berlin Heidelberg, New York, 2005.

[5] S. Theodoridis, and K. Koutroumbas, Pattern Recognition. 2nd ed. Elsevier Academic Press, Amsterdam, 2003.

[6] P. Arabie, L. J. Hubert, and G. De Soete, Clustering and Classification. River Edge. World Scientific Publishing, Singapore, 1996.

[7] M. Halkidi, and M. Vazirgiannis, “Clustering Validity Assessment: Finding the optimal partitioning of a data set,” presented at the IEEE International Conference on Data Mining, San Jose, California, USA, pp. 187–194, 29 November - 2 December, 2001.

[8] J. C. Dunn “Well Separated Clusters and Optimal Fuzzy Partitions,” Journal of Cybernetics, vol. 4, pp.95–104, 1974.

[9] R. B. Calinski, and J. Harabasz, “Adendrite Method for Cluster Analysis,” Commun. Statistics - Theory and Methods, vol. 3, no. 1, pp. 1–27, 1974.

[10] D. L. Davies, and D.W. Bouldin, “A cluster separation measure,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 1, no. 4, pp. 224– 227, 1979.

[11] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, “Validity index for crisp and fuzzy clusters,” Pattern Recognition, vol. 37, no. 3, pp. 487–501, March 2004.

[12] C. H. Chou, M. C. Su, and E. Lai, “A new cluster validity measure and its application to image compression,” Pattern Analysis and Applications, vol. 7, no. 2, pp. 205–220, July 2004.

[13] T. Zhang, R. Ramakrishnman, and M. Linvy, “BIRCH: An efficient method for very large databases,” Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, June 4-6, 1996.

[14] G. Hammerly, and C. Elkan, “Alternatives to the k-means algorithm that find better clusterings,” Proc. ACM on Information and Knowledge Management, pp. 600– 607, November 2002.

[15] P.S. Bradley, and U.M. Fayyad, “Refining Initial Points for K-Means Clustering,” ICML 1998, pp. 91–99, January 1998.

[16] W. Wang, J. Yang, and R. Muntz, “STING: A statistical Information grid approach to spatial data mining,” VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, Athens, Greece, August 25-29, 1997.

[17] P. Berkhin, “Survey of Clustering Data Mining Techniques,” Accrue Software, Technical Report, 2002.

[18] J. Kogan, Introduction to Clustering Large and High-Dimensional Data. Cambridge University Press, New York, 2007.

[19] J. Kogan, M. Teboulle, and C. Nicholas, Grouping Multidimensional Data. Springer-Verlag Berlin Heidelberg, New York, 2006.

[20] J. Valente de Oliveira, and W. Pedrycz, Advances in Fuzzy Clustering and its Applications. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2007.

[21] M. Sato-Ilic, and L. C. Jain, Innovations in Fuzzy Clustering. Springer-Verlag Berlin Heidelberg, New York, 2006.

[22] W. Pedrycz, Knowledge-Based Clustering From Data to Information Granules. John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.

[23] J. MacQueen, “Some methods for classification and analysis of multivariate bservations,” Proceedings of the Fifth Berkely Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297, 1967.

[24] A. Jain, M. Murty, and P. Flynn, “Data clustering: A Review,” ACM Computing Surveys, vol. 31, no. 3, pp. 264–323, September 1999.

[25] S. J. Redmond and C. Heneghan, “A method for initialising the K-means clustering algorithm using kd-trees,” Pattern Recognition Letters, vol. 28, no. 16, pp. 965-973, 2007.

[26] K. Mumtaz and K. Duraiswamy, “A Novel Density based improved k-means Clustering Algorithm – Dbkmeans,” International Journal on Computer Science and Engineering, vol. 2, no. 2, pp. 213-218, 2010.

[27] M. Ester, H.-P Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise,” Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, pp. 226–231, 1996.

[28] R. Storn, and K. Price, “Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces,” Technical Report TR-95- 012, ICSI, March 1995.

[29] R. Storn, and K. Price, “Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.

[30] J. A. Nelder, and R. Mead, “A simplex method for function minimization,” Computer Journal, vol. 7, no. 4, pp. 308–313, 1 January 1965.