International Journal of Computer Engineering and Information Technology

VOL. 11, NO. 9, September 2019, 197–203
Available online at: www.ijceit.org
E-ISSN 2412-8856 (Online)

An Improved Hybrid Evolutionary Clustering Algorithm to Mitigate Empty Clustering Problem

Y. A. Joarder[1], Feroza Naznin[2], Md Abdul Awal[3] and Md Zahidul Islam[4]

[1] Department of Computer Science and Engineering (CSE), World University of Bangladesh (WUB), Dhaka, Bangladesh

[2] Department of Computer Science and Engineering (CSE), Green University of Bangladesh (GUB), Dhaka, Bangladesh

[3] Department of Computer Science and Engineering (CSE), International University of Business Agriculture & Technology (IUBAT), Dhaka, Bangladesh

[4] Department of Information and Communication Technology (ICT), Islamic University (IU), Kushtia, Bangladesh

[1]yajoarder@gmail.com, [2]liaiceiu@gmail.com, [3]bongabdo@gmail.com, [4]zahidimage@gmail.com

ABSTRACT

Clustering algorithms try to get groups or clusters of data points that belong together. The main aim of this research is to improve the K-MEANS' clustering quality by eliminating empty clustering issue using the proposed hybrid partitioning algorithm titled Improved Hybrid Evolutionary Clustering with Empty Clustering Solution (IH(EC)^{2}S) and to do comparison of advanced experimental results among three best performing clustering algorithms: K-MEANS, H(EC)^{2}S and IH(EC)^{2}S respectively. Though, K-MEANS converges fairly quickly, achieving a decent solution is not guaranteed. The clustering quality is very dependent on the choice of the initial centroid selection; once the number of clusters increases, it starts to suffer from Empty Clustering issue. We have improved a hybrid partitioning algorithm that was proposed previously to remove empty clustering problem. Our proposed algorithm has employed Scalable K-MEANS++ algorithm in place of K-MEANS algorithm as it is efficient than the former one. Firstly, it clusters the whole data set. Secondly, it detects the empty cluster. Finally, it removes the empty clustering issue. Our analysis portion justifies the usages of Scalable K-MEANS++ as the best algorithm among all the clustering algorithms in terms of both performance and time complexity. Also, we have shown that IH(EC)^{2}S gives the better performance than both K-MEANS and H(EC)^{2}S.

Keywords: *Hybrid Partitioning Algorithm, Big Data, Data Clustering, K-MEANS, Evolutionary Algorithm, Scalable K-MEANS++, Cuckoo Search Algorithm, Enhanced Firework Algorithm.*

1. INTRODUCTION

Clustering is an unsupervised classification mechanism wherever a set of patterns (data), sometimes multidimensional, is classified into groups (clusters) such that members of one cluster are consistent with a predefined criterion. Clustering of a set forms a partition of its elements chosen to reduce some measure of difference between members of the same cluster. In numerous fields such as data mining, pattern recognition, learning theory and so on, clustering algorithms are typically helpful. Discovering meaningful clusters in data is one of the foremost vital aims of unsupervised learning. This research focuses on improving a previously proposed algorithm for removing the empty clusters. The new improved version of the algorithm is titled Improved Hybrid Evolutionary Clustering with Empty Clustering Solution (IH(EC)^{2}S). Typically, any specific data set does not have an unambiguously correct cluster, and the desired clustering might depend upon the actual application. Some samples of clustering applications are: to cluster connected genes from gene expression data to assist elucidate gene functions, to cluster news stories by topic to automatically organize online news feeds, and to cluster images of celestial objects so as to identify different classes of quasars and dwarfs [1]. Moreover, there has been a good deal of previous works on different strategies for clustering data, as well as hierarchical clustering, spectral clustering, K-MEANS clustering, and mixture modeling. Probably, the name of the most commonly-used clustering algorithm is K-MEANS. Between each data and its nearest cluster's center, it is the most effective. The K-MEANS finds a locally optimal solution by minimizing a distance measured for relatively smaller data sets. Many parallel versions of the K-MEANS algorithm use the fundamental K-MEANS

at their core. Besides, variety of stochastic clustering algorithms are being created using the fundamental K-MEANS or a number of its variations. Fairly often these algorithms are based on simulated annealing or genetic algorithm while these methods have been widely used in practice, many suffer from some serious limitations. For example, many of these methods do not provide a coherent way to predict the probability or cluster assignments of new data points. Here, we have worked towards improving a previously proposed hybrid partitioning algorithm: $H(EC)^2S$ by implementing our newly proposed algorithm of the same category: $IH(EC)^2S$, which overcomes K-MEANS Empty Clustering problem more efficiently than the previous one.

The K-MEANS clustering algorithm faces three major drawbacks. The first drawback is to get non-optimal solutions. As the algorithmic program is greedy in nature, it is expected to converge on a regionally optimal solution solely and not on the global optimal solution, in general. This drawback is partly resolved by applying the K-MEANS during a stochastic framework like Simulated Annealing (SA), Genetic Algorithm (GA) and so on. The second drawback is that of empty cluster generation. This drawback is additionally referred to as the singularity drawback. Singularity in clustering is obtained when one or a lot of clusters become empty. The third drawback is inefficient data clustering problem where the number of data is much lower than other clusters. All of these issues are caused by unhealthy initialization.

Here, we have presented an improved version of hybrid partitioning algorithm that eliminates the problem of empty clusters (with some exceptions) of K-MEANS algorithm. At first, the proposed algorithm does clustering of the whole data set. After that, it detects the empty clusters. Lastly, it removes the empty clustering issue. The proposed algorithm is found to be working very satisfactorily, with some conditional exceptions which are extremely rare in practice.

Contributions of this research are:

• improving the hybrid partitioning algorithm for removing empty clusters by applying Scalable K-MEANS++ algorithm [2] in place of K-MEANS algorithm in terms of running time.
• justifying the usages of Scalable K-MEANS++ as the clustering algorithm including the best running time with proper analysis.
• Showing the performance analysis among K-MEANS, $H(EC)^2S$ and our proposed improved version: $IH(EC)^2S$.

The rest of this paper is as follows. After this section I: brief Introduction, we have discussed Literature Review in section II. After that, we have provided Clustering Algorithms and Data Analysis in section III; in section IV, our proposed Partitioning Algorithm is described. In addition, Experimental Result and Discussion is described in section V. Lastly, we have discussed the Conclusion in section VI.

2. LITERATURE REVIEW

K-MEANS [3] is the most generally used algorithm for clustering data because of its pertinence and simplicity. There are some studies have enforced on optimizing completely different objectives of K-MEANS algorithm such as Euclidian k-medians [4] and Geometric k-center [5]. Minimization of the sum of distances to the closest center is the goal for Euclidean k-medians, and minimization of the maximum distance from each point to its nearest center is the one for geometric k-center version. Another research has done to hunt a better objective function of K-MEANS [6]. Although, there are completely different versions of K-MEANS that may have benefits, parallelization of algorithm in a single machine resulted in significant performance improvement [7]. Achieving the parallelization over multiple machines results in even higher improvements.

The framework Map Reduce [8] provides significant enhancements of scalable algorithms. There has been many studies for clustering large scale data on distributed systems in parallel on Hadoop [9]. One such approach is HaLoop [10] that is a modified version of the Hadoop creating the task hardware loop-aware and by adding varied caching mechanisms. Another approach to cluster data in a distributed system was using Apache Mahout Library [11]. Moreover, clustering of huge data can be done in cloud conjointly. The tests were running on Amazon EC2 instances and also the comparisons were created to realize the gain between the nodes [12]. Esteves et al. created comparisons over K-MEANS and Fuzzy C-MEANS for clustering of Wikipedia massive scale data set [13]. In addition, fireworks algorithm, that was inspired by observing reworks explosion, is a recent meta-heuristic that has been projected by Tan et al [14]. Authors showed that it outperforms standard PSO and clonal PSO in experiments. Advanced fireworks algorithm is the improved version of the fireworks algorithm [15]. Moreover, another meta-heuristic named cuckoo search has been projected, which was galvanized by obligate brood parasitic behavior of some cuckoo species together with the levy flight behavior of some birds and fruit flies.

A refinement approach is projected, where there is beginning of variety of initial samples of the dataset, variety of sets of center vectors [16]. These center vectors then go through a refinement stage to get a set of these so called good starting vectors. In [17], a genetically guided KMEANS has been planned wherever chance of generation of empty clusters is expounded within the mutation stage. Many k-d-tree based methods are mostly found in [18] and [19]. Another approach to initialize cluster centers based on values for every attribute of the data set has been planned

on [20]. These ways are time consuming and may not be applicable by keeping the K-MEANS in inherently straightforward structure. There is additionally another no drawback with K-MEANS algorithm: when the number of cluster formation increases, possibility of getting empty clusters (locations that do not have any data points related to the clusters) additionally increases at every iteration. This becomes an ineluctable issue when k >> 1. This issue was not addressed completely in most of the studies.

3. CLUSTERING ALGORITHMS AND DATA ANALYSIS

Clustering algorithms have emerged as another powerful meta-learning tool to accurately analyze the huge volume of data generated by trendy applications. Especially, their main goal is to categorize data into clusters in such a way that, objects are grouped within the same cluster when they are similar in line with specific metrics. There is a massive body of knowledge within the space of clustering and there are attempts to research and categorize them for a bigger range of applications. However, one among key problem in using clustering algorithms for giant data that causes confusion amongst practitioners is that the lack of accord within the definition of their properties also as a lack of formal categorization.

In the current digital era, in keeping with huge progress and development of the online world technologies like big and powerful data servers, we have been facing a huge volume of knowledge and data day by day from many alternative resources and services that were not available for human beings simply many decades ago. Huge quantities of data are being created and concerning individuals, things and their interactions. Numerous teams argue concerning the potential advantages and prices of analyzing information from Google, Facebook, Twitter, Wikipedia and each area wherever massive teams of individuals leave digital traces and deposit data. This data comes from completely different available online resources and services that have been established to serve their customers. In addition, services and resources like cloud storages, sensor networks, social networks and so on, turn out to have massive volume of data and conjointly have to be compelled to manage and recycle that data or so mean analytical aspects of that data. Though this huge volume of data can be very helpful for individuals and corporations; it can be problematic as limitations further. They need big storages and this volume makes operations like analytical operations, process operations, retrieval operations terribly tough and vastly time well. Therefore, a massive volume of big data has its own overwhelming. A technique to beat these tough issues is to possess massive information clustered in association with nursing exceedingly compact format that is still an informative version of the whole data. Such clustering techniques aim to supply a decent quality of clusters. Therefore, they might vastly profit everybody from normal users to researchers and people within the corporate world, as they could offer an efficient tool to deal with massive data like essential systems to find cyber-attacks. There are mainly five types of clustering algorithm:

1. Partitioning-Based
2. Model-Based
3. Grid-Based
4. Density-Based
5. Hierarchical-Based

Below an overview of taxonomy clustering algorithms is presented in Fig.1.

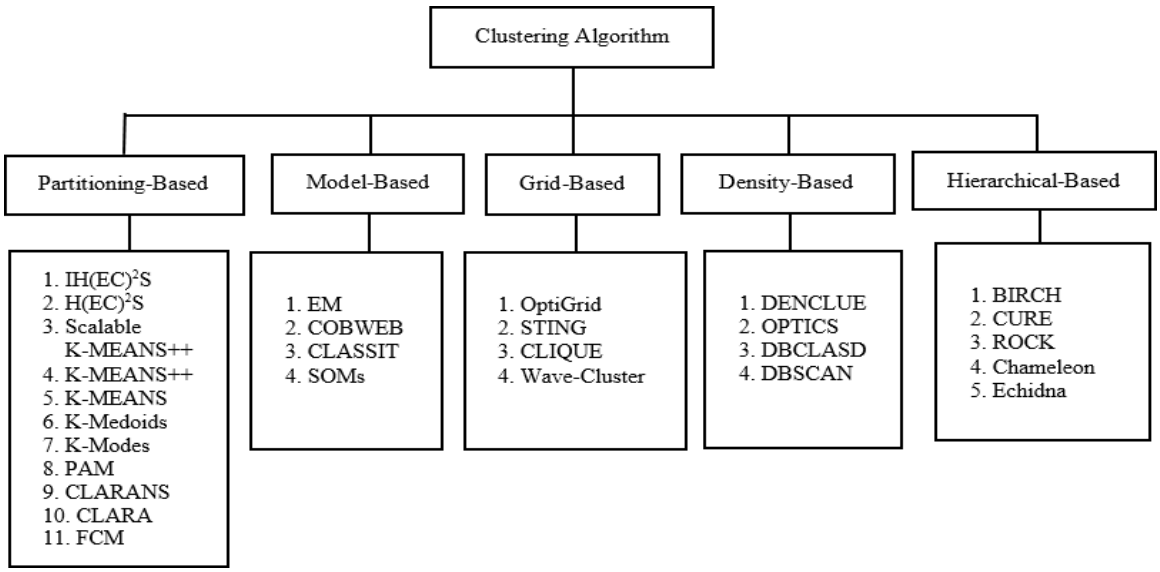

Fig. 1. Taxonomy of Clustering Algorithms.

Table I: Compliance Summary of Clustering Algorithms

Clustering Algorithms	Internal Validity	External Validity	Efficiency Problem	Stability	Scalability
IH(EC)^{2}S	Partially	Yes	Suffer From	Suffer From	Low
H(EC)^{2}S	Partially	Yes	Suffer From	Suffer From	Low
Scalable K-MEANS++	Partially	Yes	Suffer From	Suffer From	Low
K-MEANS++	Partially	Yes	Suffer From	Suffer From	Low
K-MEANS	Partially	Yes	Suffer From	Suffer From	Low
EM	Partially	Yes	Suffer From	Suffer From	Low
OptiGrid	Yes	No	Yes	Suffer From	High
DENCLUE	Yes	No	Yes	Suffer From	High
BIRCH	Suffer From	No	Yes	Suffer From	High

We can say that Scalable K-MEANS++, K-MEANS and H(EC)^{2}S are partitioning-based clustering algorithms from Table I. In addition, based on empirical evaluation metrics and compliance summary of clustering algorithms, we can say that among all the clustering algorithm, Scalable K-MEANS++ perform better than others except IH(EC)^{2}S and H(EC)^{2}S. By using Scalable K-MEANS++, it is easy to calculate clustering center and improve poor clustering quality of K-MEANS algorithm. It creates possible functions in the basis of distance. In addition, Scalable K-MEANS++ algorithm works quickly than that of K-MEANS algorithm and save time of clustering.

4. PROPOSED HYBRID PORTIONING ALGORITHM

Here, we have discussed all about our proposed algorithm description step by step. Our proposed model titled Improved Hybrid Evolutionary Clustering with Empty Clustering Solution (IH(EC)^{2}S) consists of three different algorithmic parts: A) RC part, B) EFC part and C) CSC part.

Algorithm 1:IH(EC)^2*(Improved Hybrid Evolutionary Clustering Algorithm Empty Clustering Solution)*

1. **Run** Representative Construction (RC) Part
2. **For** *t =1* to *itermax* **do**

{

1. **Run** Enhanced Firework Algorithm for Clustering (EFC) Part
2. **Run** Cuckoo Search Algorithm for Clustering (CSC) Part

}

3. **Run** Scalable K-MEANS++ with the best firework

A. Representative Construction (RC) Part

Representative Construction (RC) Part selects the representative values for each data sample and eliminates outliers. To do this, it first find distinct data points from discrete values of each dimension and map them to specific representative value. They are less than certain threshold according to outliers [21].

B. Enhanced Fireworks Algorithm for Clustering (EFC) Part

Enhanced Fireworks Algorithm for Clustering (EFC) Part selects the good initial points among the selected representatives at the previous part. It detects the empty clusters by searching the representative without any data points [21].

C. Cuckoo Search Algorithm for Clustering (CSC) Part

Cuckoo Search Algorithm for Clustering (CSC) part keeps updating the new fireworks to pass that to next generation and eliminates the clusters that marked empty by Enhanced Fireworks Algorithm for Clustering Part [21].

5. EXPERIMENTAL RESULT AND DISCUSSION

A. System Implementation Environment

Data science, analytics, machine learning, big data are all acquainted terms in todays' tech headlines, however, they can appear discouraging, opaque or just simply not possible. We can dive into what information science consists of and how we can use Python to perform data analysis for us. Data science is a massive field covering everything from data assortment, cleaning, standardization, analysis, visualization and reporting. There are many alternative positions, firms and fields that touch data science.

The tools that have been used for the experimental procedure are:

- VM WARE
- Data Sets (Irish, Crude Oil and so on)
- Canopy IDE
- NoSQL
- Hadoop
- Data Science Library
- Python Library

• Python Framework

The overall system design is presented in Fig. 2. Here, we have used VM WARE for creating our Virtual Big Data Environment. Python has emerged over the past few years as a front-runner in data science programming whereas there are still scores of folks using R, SPSS, Julia or many various common languages. Python growing quality within the field is clear within the growth of its data science libraries. That is why, we have used the Python programming with canopy IDE for this research to improve hybrid partitioning clustering implementation. There have been used different types of Python Libraries such as: Math Plot Pyplot and so on. IPython has been used as a Python Framework. In addition, distributed computing has been used for system implementation. The main server has been connected with Big Data storage (Data Warehouse), Canopy IDE and Hadoop respectively. NoSQL (MongoDB) has been used as a Standard Query Language (SQL) in Big Data Storage where different types of data sets are available such as: Irish Data, Spherical 4 3, Circular 5 2, Elliptical 10 2, Circular 6 2, Color Moments, Crude Oil and Breast Cancer and so on. We have used Hadoop for creating distributed data management system where Spark is used as for Map Reducing Framework. Scikit has been used as a data science library. N sub servers for N clusters have been used here as well.

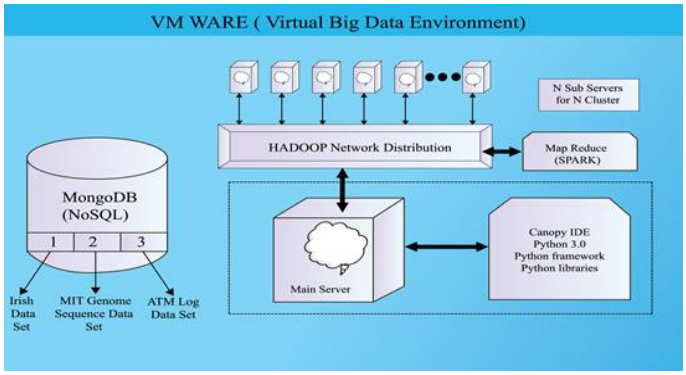

Fig. 2. Design of the system implementation environment

B. Result Discussion on Used Data Sets

In our result analysis part, we have compared our result with the previous study [21], where the authors of the previous study compared the performance of H(EC)^{2}S and K-MEANS algorithm specifically. We have added our result with IH(EC)^{2}S algorithm with the result analysis done by previous study and have shown that in Table II, Fig.3., Fig. 4. and Fig. 5. respectively.

TABLE II
ITERATION NUMBERS AND METRIC VALUES OF USED DATA SETS

Data Sets	K-MEANS		H(EC)^{2}S		IH(EC)^{2}S	
	Iteration Number	Metric Value	Iteration Number	Metric Value	Iteration Number	Metric Value
Iris	6	98.210	11	98.205	9	98.201
Spherical 4 3	5	751.960	9	751.941	6	751.935
Circular 5 2	5	331.531	7	331.527	6	331.521
Elliptical 10 2	13	949.391	11	951.347	9	697.711
Circular 6 2	5	374.541	6	374.542	6	374,537
Color Moments	153	133185.285	97	132942.563	72	132942.559
Crude Oil	15	281.745	11	281.744	9	281.742
Breast Cance	4	2988.961	7	2988.963	5	2988.958

It is noticeable that, from the TABLE II which shows the iteration numbers and metric values of used data sets. The iteration number reduces for the higher metric value by using H(EC)^{2}S than K-MEANS on the different types of data sets and our proposed method reduces the number of iterations further. For example: in Color Moments data sets the iteration number is 153 by implementing K-MEANS while the number is 97 by implementing H(EC)^{2}S , where our method IH(EC)^{2}S required. Only 72 iterations are lesser than the both. In both of the cases, the Metric Value is higher than other data sets in the given TABLE II. Similar situation occurs for Elliptical 10 2 and Crude Oil datasets. For Elliptical 10 2 dataset H(EC)^{2}S has used 9 iterations in comparison to 11 iterations for H(EC)^{2}S and 13 iterations for K-MEANS. For Crude Oil dataset IH(EC)^{2}S has used 9 iterations in comparison to 11 iterations for H(EC)^{2}S and 15 iterations for K-MEANS. However, the opposite is right for lower metric value in most of the cases such as in Iris data sets, the iteration number is 6 by implementing K-MEANS whereas the number is higher by 5 by implementing H(EC)^{2}S . IH(EC)^{2}S metric gives better performance than the H(EC)^{2}S even in these cases. However, for lower metric value IH(EC)^{2}S method is still lagged behind the K-MEANS algorithm. In Iris data set, the metric value is lower than other data sets in the given TABLE II.

C. Performance Analysis

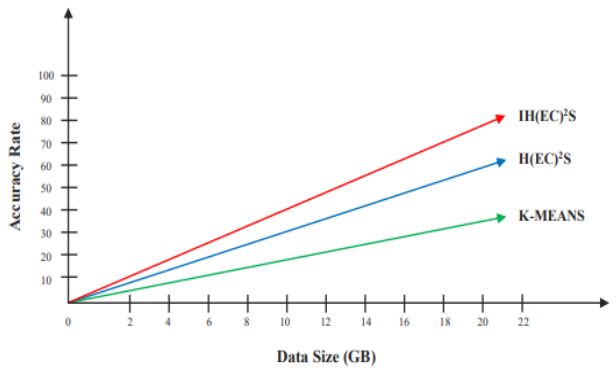

Fig. 3. Accuracy Rate Analysis among K-MEANS, H(EC)²S and IH(EC)²S.

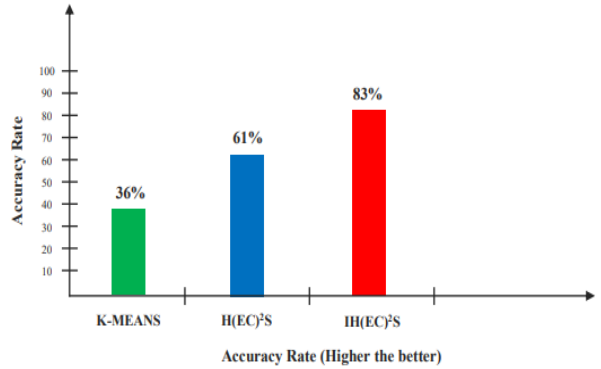

Fig. 4. Percentages of Accuracy Rate among K-MEANS, H(EC)²S and IH(EC)²S.

The accuracy rate of the output result between K-MEANS, H(EC)²S and IH(EC)²S represent in respect of Data size and Accuracy Rate in Fig.3. The percentage of accuracy rate of the output result of IH(EC)²S, is much higher than that of K-MEANS and H(EC)²S on the big data size. For example, when the data size is around 1 GB, the performance of IH(EC)²S is nearly equal to both the H(EC)²S and K-MEANS algorithm. However, the accuracy gradually increases as the volume of the data size increases and for data size above 20 GB. From both Fig. 3. and Fig. 4., We have achieved accuracy about 83%, 61% and 36% respectively. H(EC)²S and K-MEANS algorithm. However the accuracy gradually increases as the volume of the data size increases and for data size above 20 GB we have achieved accuracy about 83%, 61% and 36% respectively.

D. Time Complexity Analysis

The comparison of time complexity between IH(EC)²S, H(EC)²S and K-MEANS with respect to Data size (GB) and Time shows in Fig. 5.

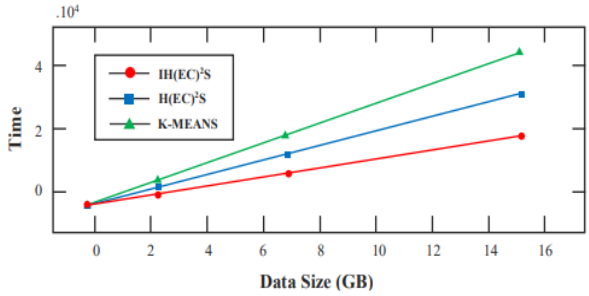

Fig. 5. Time complexity comparison among K-MEANS, H(EC)²S and IH(EC)²S.

The original K-MEANS clustering algorithm has much higher time complexity than that of IH(EC)²S. In addition, IH(EC)²S is also better than H(EC)²S in respect to time complexity. On the similar note, as data volume increases, the gap among the three algorithm for time also increases. For example: when data size is around 2 GB, time complexity for the three algorithms are very close and they require nearly same amount of time. However, when data size is near 20, time complexity between IH(EC)²S to K-MEANS is almost twice and between IH(EC)²S to H(EC)²S almost one and half times.

6. CONCLUSION

Improving the performance of K-MEANS algorithm by removing empty clusters could be a significant research innovation within the domain of Data Science. Careful selection of centroids can remove the empty clustering problem. Our research has tried to improve a proposed method which attempted to ameliorate clustering by removing empty clusters. Moreover, our proposed improved hybrid partitioning algorithm has run time advantage over the previously proposed method H(EC)²S as we have used Scalable K-MEANS++ in place of K-MEANS for the convergence with best fireworks. Moreover, Scalable K-MEANS++ eliminates the poor clustering choices of K-MEANS. For this we have shown with analysis that partitioning algorithms work better for clustering and among them Scalable K-MEANS++ is the best on the basis of its performance. We have implemented our proposed method and it shows that iteration number decreases especially for the large data sets than K-MEANS and H(EC)²S. In future, we are wishing to develop several trendy and advanced clustering frameworks. Moreover, we will investigate the question: how can the most suitable parameter settings be found for each clustering algorithm?

REFERENCES

[1] D. Pelleg, A. Moore, Accelerating exact k-means algorithms with geometric reasoning, Tech. rep., CARNEGIE-MELLON UNIV PITTS- BURGH PA SCHOOL OF COMPUTER SCIENCE (2000).
[2] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, S. Vassilvitskii, Scalable k-means++, Proceedings of the VLDB Endowment 5 (7) (2012) 622–633.
[3] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley sympo- sium on mathematical statistics and probability, Vol. 1, Oakland, CA, USA, 1967, pp. 281–297.
[4] S. G. Kolliopoulos, S. Rao, A nearly linear-time approximation scheme for the euclidean k-median problem, SIAM Journal on Computing 37(3) (2007)757–782.
[5] P. K. Agarwal, C. M. Procopiuc, Exact and approximation algorithms forclustering, Algorithmica33 (2)(2002)201–226.
[6] J. A. Hartigan, M. A. Wong, Algorithm as 136: A k-means clustering algorithm, Journal of the Royal Statistical Society. Series C (Applied Statistics) 28 (1) (1979) 100–108.
[7] X. Li, Parallel algorithms for hierarchical clustering and cluster validity, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (11) (1990) 1088–1092.
[8] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Communications of the ACM 51 (1) (2008) 107–113.
[9] B. O'Sullivan, Mercurial: The Definitive Guide: The Definitive Guide, " O'Reilly Media, Inc.",2009.
[10] Y. Bu, B. Howe, M. Balazinska, M. D. Ernst, Haloop: efficient iterative data processing on large clusters, Proceedings of the VLDB Endowment 3 (1-2) (2010) 285–296.
[11] S. Owen, R. Anil, T. Dunning, E. Friedman, Mahout in action: Manning Shelter Island.
[12] R. M. Esteves, R. Pais, C. Rong, K-means clustering in the cloud–a mahout test, in: 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications, IEEE, 2011, pp. 514–519.
[13] C. Rong, et al., Using mahout for clustering wikipedia's latest articles: A comparison between k-means and fuzzy c-means in the cloud, in: 2011 IEEE Third International Conference on Cloud Computing Technology and Science, IEEE, 2011, pp. 565–569.
[14] Y. Tan, Y. Zhu, Fireworks algorithm for optimization, in: International conference in swarm intelligence, Springer, 2010, pp. 355–364.
[15] S. Zheng, A. Janecek, Y. Tan, Enhanced fireworks algorithm, in: 2013 IEEE congress on evolutionary computation, IEEE, 2013, pp. 2069– 2077.
[16] P. S. Bradley, U. M. Fayyad, Refining initial points for k-means clustering., in: ICML, Vol. 98, Citeseer, 1998, pp. 91–99.
[17] F.-x. Wu, Genetic weighted k-means algorithm for clustering large-scale gene expression data, BMC bioinformatics 9 (6) (2008) S12.
[18] A. Likas, N. Vlassis, J. J. Verbeek, The global k-means clustering algorithm, Pattern recognition 36 (2) (2003) 451–461.
[19] S. Deelers, S. Auwatanamongkol, Enhancing k-means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance, International Journal of Computer Science 2 (4) (2007)247–252.
[20] S. S. Khan, A. Ahmad, Cluster center initialization algorithm for k- means clustering, Pattern recognition letters 25(11)(2004)1293–1302.
[21] J. Karimov, M. Ozbayoglu, High quality clustering of big data and solvingempty-clusteringproblemwithanevolutionaryhybridalgorithm, in: 2015 IEEE International Conference on Big Data (Big Data), IEEE, 2015, pp.1473–1478.