Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7683991

STUDY ON DEVELOPMENT AND CHALLENGES OF CLUSTERING ALGORITHMS

Dinesh Bhardwaj and Dr. Sonawane Vijay Ramnath
Department of Computer Science & Engineering, Dr. A. P. J. Abdul Kalam University, Indore (M.P.) – 452010
Corresponding Author Email: dkbh28@gmail.com

Abstract:
This paper examines six distinct clustering methods: k-means clustering, hierarchical clustering, DBS can cluster, density-based clustering, optical flow, and EM algorithm. WEKA, a clustering tool, is used to carry out the implementation and analysis of these clustering techniques. Six different methods' results are shown and compared. Retrieving data from scientific and technical literature via R-tree indexing, our method employs an enhanced k-mean clustering algorithm to build a clustering model. The experiments conducted on university science and technology literature datasets demonstrate the effectiveness of the approach described in this paper. Clustering is a well-known, fundamental data mining task that is used to extract information. However, many researchers have developed and provided a wide variety of clustering algorithms to accommodate the adapted applications for the various domains. Because of this, it is challenging for researchers and practitioners to keep up with the progress being made in clustering algorithm development.

Keyword – Data Clustering, K-Means Clustering, Hierarchical Clustering, DB Scan Clustering, Density Based Clustering, OPTICS, EM Algorithm

1. INTRODUCTION

CLUSTERING is a data-mining method for organizing data into clusters based on their similarities and differences. The goal of clustering, and every other issue of this kind, is to discover patterns in data that has not been labeled in any way. To cluster is to organize things into groups where the members are similar in some manner. In this sense, we might define a cluster as a grouping of items that share similarities among themselves but contrast with those of other clusters. Clustering refers to the unsupervised process of grouping similar patterns (observations, data objects, or feature vectors) into larger categories (clusters).

Using a method called data clustering, similar pieces of information are grouped together. The goal of a clustering method is to divide a dataset into subsets with more internal similarity than external similarity. Most of the data obtained in many cases also seems to have some intrinsic features that allow for natural classifications. In addition to their widespread use in data organization and categorization, clustering algorithms also find application in data compression and model building.

Discovering relevant knowledge in data is another motivation for clustering. Finding these groupings or attempting to categorize the data is not an easy task for or three dimensions at

most. When no preexisting categories adequately describe a user's needs, data clusters are developed to fill the void. Data subjects can be clustered together temporarily.

Disk structure:

(A) Track

(B) geometrical sector

(C) Track sector

(D) Cluster

The goal of each of the described clustering methods is to identify representative centers for each cluster. A cluster center is a method to identify the core of each cluster, which is useful for establishing which cluster an input vector belongs to by comparing it to all of the cluster centers and identifying the one with the highest similarity score. Knowing the target number of clusters in advance is a prerequisite for several clustering methods. In such situation, the algorithm will attempt to divide the data into that many distinct groups. Methods like K-means and Fuzzy C-means clustering fall within this category.

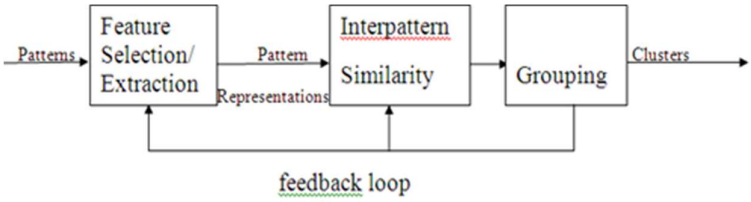

Figure 1: Stages in clustering

There are several methods for carrying out the grouping stage. The final clustering (or clusterings) might be hard (a strict categorization of the data) or fuzzy (where each pattern has a variable degree of membership in each of the output clusters).

2. LITERATURE REVIEW

Taeho Jo (2020) This article suggests a strategy for text clustering based on a variant of the Agglomerative Hierarchical Clustering (AHC) algorithm that groups string vectors rather than numerical vectors. Previous studies found promising results when using string vector based methods to text clustering, and a synergy effect between text clustering and the word clustering is predicted by merging them; these facts motivate the current study. In this study, we introduce the concept of "semantic similarity" as an operation performed on string vectors and adapt the AHC algorithm to use this similarity metric in its text clustering process. Empirical evidence supports the superiority of the proposed AHC algorithm in grouping sentences in news articles and views. More complex machine learning algorithms will need the addition of new string

vector operations, hence it is necessary to describe and characterize these operations theoretically.

Laith Abualigah (2022) The purpose of text document clustering is to classify or categorize textual content into distinct groups. Due to the massive amounts of daily data from the Web, it attracted a lot of interest. Meta-Heuristic (MH) methods have been widely used in the previous decade to address clustering issues. To address this issue, the authors present the Arithmetic Optimization Algorithm, a robust implementation of the recently developed MH algorithm. The AOA is based on the four arithmetic operations (multiplication, subtraction, addition, and division) that underlie all mathematics. Although the AOA did well on a number of global problems, it gets stuck in low-quality local minima when dealing with problems that are both complex and high-dimensional. As a result, this paper suggests a refined version of AOA to solve the document clustering issue in text. To address the shortcomings of the original AOA, the Improved AOA (IAOA) integrates Opposition-based learning (OBL) and Levy flight distribution (LFD) into the algorithm. The IAOA is evaluated as a global optimization algorithm by comparing it extensively to preexisting optimization algorithms and testing it on a variety of UCI datasets for text clustering problems. The suggested IAOA is shown to be superior to other optimization techniques in a variety of experiments. In addition, twenty-one state-of-the-art methods are used to compare the proposed IAOA to 31 benchmark text datasets, demonstrating the superiority of the proposed IAOA.

Chunhua Tang (2021) It is difficult to establish parameters for most density-based clustering methods, and these algorithms are slow and inefficient, bad at recognizing noise, and unable to properly cluster data sets of varying densities. In this study, we propose FOP-OPTICS (Finding of the Ordering Peaks Based on OPTICS) as a significant enhancement over OPTICS that may be used to address these issues (Ordering Points To Identify the Clustering Structure). Using the OPTICS-generated Augmented Cluster-Ordering, the suggested technique locates the demarcation point (DP) and use the DP's reachability-distance as the neighborhood eps radius for the related cluster. It fixes a problem that plagues the vast majority of algorithms when trying to cluster data sets of varying densities. OPTICS' time complexity is reduced because to this method's ability to effectively distinguish noise via the calculation of density-mutation sites inside clusters. Based on the experimental data, it is clear that FOP-OPTICS has the best performance in terms of parameter tuning and noise recognition, as well as the lowest time complexity.

Igor Škrjanc (2022) In this research, we describe a data-driven method for dynamically classifying Twitter users into similarity groups. Clustering is performed using a Takagi-Sugeno fuzzy consequent component of order zero and a Gaussian probability density function (eGauss0). This suggests that the technique may serve as a classifier, that is, a mapping from the feature space to the class label space. The eGauss approach is highly adaptable, uses recursive computation, and, most importantly, begins its learning process with a clean slate. As new information is added, the structure merges and expands to accommodate it. One key aspect of the developing technology is its ability to tackle the Big Data challenge posed by processing data from thousands of Twitter accounts in real time. Different degrees of Twitter engagement

serve as proxies for the final clusters, which in turn generate classes of user profiles. This would allow us to categorize each user as typical, very engaged, influential, or out of the ordinary. We also compared the suggested approach to others by testing it on the Iris and Breast Cancer Wisconsin datasets. Both scenarios benefit from the high categorization rates and competitive outcomes that the suggested approach provides.

Vo Ngoc Phu (2017) Classifying emotions has important applications in many fields, including daily life, politics, commodity production, and business. Emotion classification is a complex problem that requires a timely and reliable solution. We propose a novel model for sentiment classification using big data and a parallel network architecture in this study. We propose a model for cloud-based English sentiment classification using Fuzzy C-Means (FCM) and Hadoop MAP (M) /REDUCE (R). Cloudera is a distributed, parallel database system. In a parallel network setting, our proposed model can categorize the emotions expressed in millions of English documents. The 25,000 English-language reviews in the testing data set were split evenly between positive and negative, and our model achieved an accuracy of 60.2%. There are a total of 60,000 English sentences in our training data set, of which 30,000 are considered "positive" and "negative."

3. METHOD

The clustering articles have grown in significance during the last twenty years, indicating that scholars are paying more and more attention to this issue. Using the Science Direct database and a filter for research and review articles, we divided the available literature into four broad categories: reviews and surveys; comparative studies; clustering techniques that focused on a novel algorithm; and clustering applications. Figure 1 shows that 23% of papers focused on comparing different algorithms, while 38% applied clustering to domains including image processing, speech processing, information retrieval, Web applications, and industry, all based on the [10] methodology. This methodical procedure assured the review's thoroughness, leading to the examination of many comparative studies of clustering algorithms across several application areas. Twenty of the 32 articles chosen from the literature survey were utilized for comparative analyses, and ten were used to apply the clustering approach to the relevant industry.

Based on our review of these works, we can say that many researchers have examined well-known algorithms like K-means, DBSCAN, DENCLUE, K-NN, fuzzy k-means, and SOM in great detail, discussing their merits and shortcomings and taking into account a wide range of contextual factors that may affect the selection of the most suitable clustering algorithm for a given dataset. In contrast, other studies have investigated the feasibility of providing clustering algorithms surveys based on a variety of criteria, including but not limited to: score (merits), problems solved, applicability, domain knowledge, and size of dataset, number of clusters, type of dataset, software used, time complexity, stability, and so on. Researchers have also found that the difficulties of working with large amounts of data might be a problem for clustering algorithms. They present a framework for classifying existing clustering algorithms into categories according to the 4Vs of big data, namely Volume, Variety, Velocity, and Value, and

draw conclusions about the most appropriate algorithm for various big datasets in terms of internal, external, stability, and runtime performance indices. K means, DBSCAN, agglomerative hierarchical clustering, and the SOM method are only some of the algorithms that have been studied in the context of clustering packaging and environmental risk, financial, female worker, consumer preference, industrial hygiene, and forest sector datasets. Despite the abundance of literature reviews and comparisons, there are still gaps in our understanding, such as a lack of research into the algorithms' individual characteristics and a lack of rigorous empirical analysis that would allow us to determine which algorithm is superior for any given type of dataset.

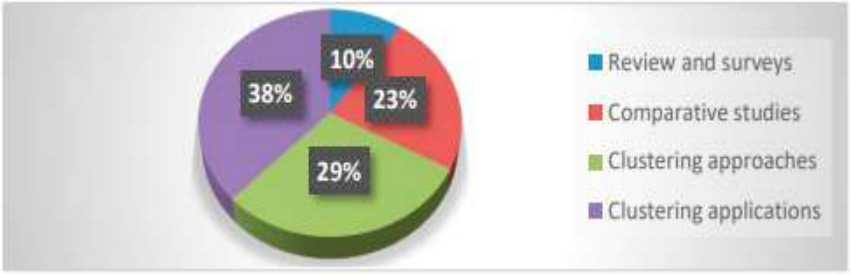

Figure 2: Classification of clustering publications from 1993 to 2017

Furthermore, there is no paper that deals with different algorithms properly evaluated and compared on real industrial datasets, with the exception of two studies that compare DBSCAN and Keans for financial datasets and agglomerative hierarchical clustering and SOM for packaging modularization datasets. As a result, the problem of how to best find the optimal clusters for sparse industrial datasets and provide a comprehensive summary of those approaches remains unanswered. In light of these considerations, the next part provides a classification framework for current algorithms, with the goal of selecting candidate clustering algorithms for appropriate evaluation by comparing their benefits and downsides.

3. RESULT AND DISCUSSION

Experiment Preparation. Taking into account the calculation process, the text similarity is larger and the dimensionality generated is higher; therefore, the text category, title, keyword, and keyword are analyzed when clustering is performed to reduce the dimensionality generated by the text during the calculation, while the experiment is conducted using the method proposed in this paper; the experimental environment is as follows: System requirements for the experiment were a Windows 10 64-bit OS, an Intel Core i5-5200 2.2 GHz CPU, and 6GB of RAM. * With a total of 19637 documents across 20 categories, including agriculture, art, military, computer, economy, education, environment, and medicine, the dataset was derived from the text classification language library provided by the natural language processing group at a university's database center. Of the 2627 texts included in the dataset, 1839 were used for training, while the remaining 788 were used for testing. Index metrics like P (precision), R (recall), and F1-score are often used to measure the efficacy of document clustering (correlated with accuracy and checking completeness).

Table 1: Parameter settings.

Parameters	Values
W	100 MB
T_1, T_2	0.47 MB, 4.48 mb
N_1, N_2	5549, 12068
Q_1, Q_2	1, 4
$\alpha = 0$	100 MB
$\alpha = 1$	50 MB
$\beta = 0$	$T_1 + N_1 + Q_1$
$\beta = 1$	$T_2 + N_2 + Q_2$

Analysis of Retrieval Efficiency Based on R-Tree Clustering.

*The quantity of data plays a key role in determining the size of the R-tree. For a given data quantity T and available network bandwidth Was. Indicate the query complexity as Q and the number of queries as N. The available bandwidth, denoted by the experimental variable, and the number of tasks, denoted by, determine the values for the other parameters, which are set as shown in Table 1. Figures 3 and 4 depict the loss convergence and performance improvement achieved during the training process, respectively. The experimental results are displayed in Figure 5, and they show that when the task size is small, the retrieval time of the system deploying R-tree is slower than hash index at the beginning, but after a period of time, the speed of the system increases.

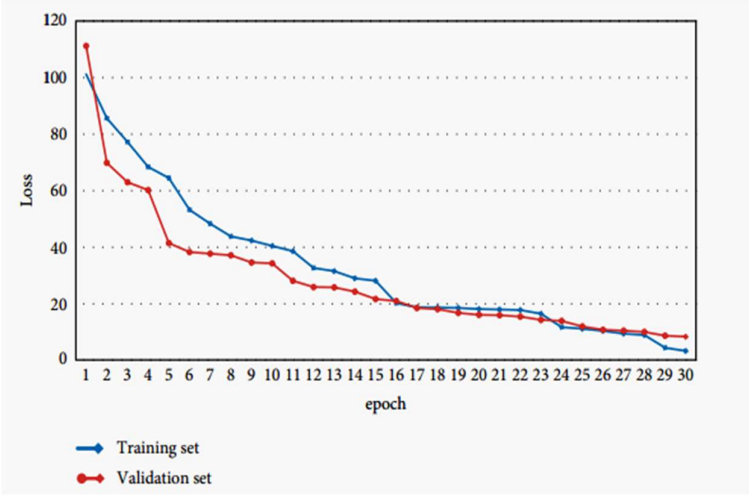

Figure 3: Lossy convergence diagram of the training process.

Since R-trees must be built by the deploying system before any retrieval can occur, and since the approach described in this study has a temporal complexity of O (nk t), where k is the number of clusters, t is the number of iterations, and n is the quantity of data, this is the expected outcome. Further, hash index has an O(1) time complexity (1). This means that R-tree index is less efficient than hash index when it comes to actual system performance. The R-tree built in this paper, however, can effectively cut down on overlap and coverage between MBRs, resulting in a more compact generated tree structure and fewer multipath queries, both of which

boost retrieval efficiency. R-retrieval tree's performance surpasses that of hash index after some time has passed while the system is operating.

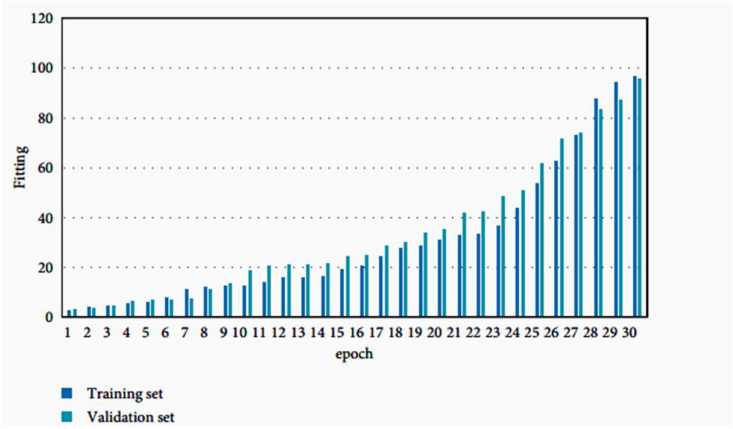

Figure 4: Performance improvement of the training process.

Figure 6 demonstrates that under high work loads, the system performance of deploying an R-tree is superior to that of hashing. In a crowded network, the benefit becomes increasingly evident. After running a few trials on a busy network, the hash index deployment system crashes and stops working, making further experiments unfeasible. The remaining procedures may still be carried out by the system installing R-tree. This is due to the fact that R-query tree's response time is affected by the number of lookups, data size, and complexity of the search pathways. Also, creating and maintaining the hash table places a heavy load on the computer's processing performance; when the quantity of data is big, this will have a negative impact on the system's performance after some time has elapsed.

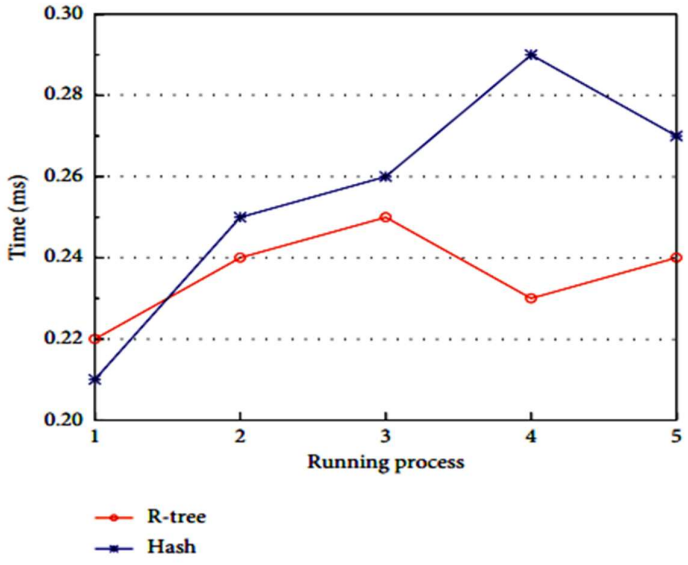

Figure 5: System performance at. $\alpha = 1$, $\beta = 0$.

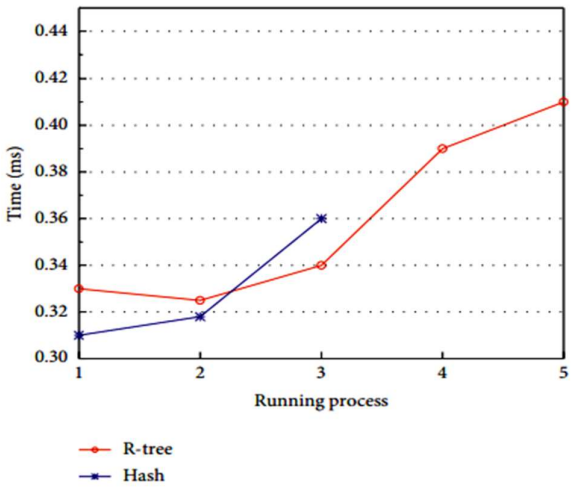

Figure 6: System performance at. $\alpha = 0$, $\beta = 1$.

Search Precision Analysis. In this work, we evaluate it in relation to the KNN and DBSCAN methods. Table 2 displays a comparison of the three classification techniques at varying values. Based on the table's experimental results, we know that the clustering effect of the KNN algorithm is greatest when P, R, and F1 are all at their highest possible values of 19; that of the DBSCAN algorithm is greatest when it reaches its highest possible value of 14; and that of the algorithm presented here is greatest when P, R, and Fl are all at their highest possible values of 17. The second stage of the experiment was run to assess the efficacy of the three distinct algorithms while pursuing the ideal values; the resulting experimental results are shown in Figure 7 and Table 3. Two sets of data, shown in Figure 7 and Table 3, compare the F1 values and time consumption of the KNN algorithm and the DBSCAN algorithm with the approach suggested in this article after clustering the data set. Figure 7 demonstrates that the F1 values achieved using the DBSCAN method are much higher than those produced using the KNN technique, and that the approach presented here improves upon both of these. Table 3 shows that the KNN technique requires the most time for clustering, and that DBSCAN requires 18 seconds more time than the approach suggested in this article.

Table 2: Comparison of retrieval accuracy with other algorithms.

Times	KNN				DBSCAN				Proposed			
	Values	*P*	R	F1	Values	*P*	R	F1	Values	*P*	R	F1
1	7	72.9	73.1	73.0	8	85.6	80.0	82.7	9	81.3	84.5	82.9
2	10	74.4	72.6	73.5	10	82.9	91.1	81.9	11	83.2	82.1	82.6
3	13	77.4	74.2	75.8	12	78.6	91.2	84.4	13	85.6	88.3	86.9
4	16	79.7	78.6	79.1	14	95.2	95.6	95.4	15	92.3	95.7	94.4
5	19	82.7	79.4	81.0	16	91.4	93.2	92.3	17	95.8	96.2	96.0
6	22	79.4	76.6	78.0	18	85.3	90.2	87.7	19	86.8	91.4	89.0
7	25	78.3	73.3	75.7	20	83.6	87.1	85.3	21	83.5	87.3	85.4
8	28	76.5	72.4	74.4	22	81.3	84.6	82.9	23	81.1	83.2	82.1
Mean		77.7	75.0	76.3		85.5	87.9	86.6		86.3	88.6	87.4

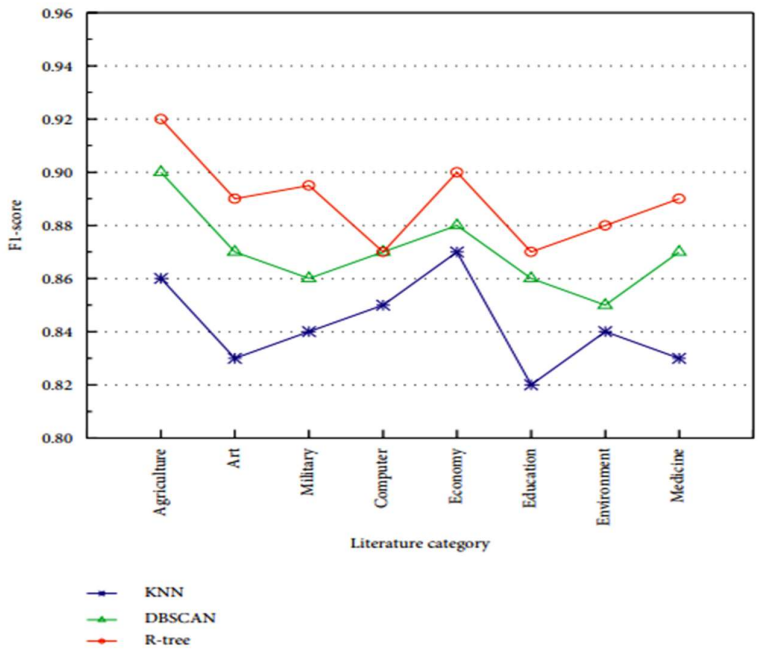

Figure 9: Comparison of F1-score of three algorithms

Table 3: Time consumed by the three algorithms to query all literature**.**

Method	Time (s)
KNN	1045
DBSCAN	893
Proposed	875

4. CONCLUSION

Because most clustering software follows the same approach when executing any algorithm, running the clustering algorithm using any software yields almost the same result even when altering any of the parameters. This study, in particular, provides a novel search approach that takes into account the peculiarities of several literary data formats and a high data volume by using R-tree indexing. The experimental findings presented in this work demonstrate a significant improvement in both efficiency and accuracy when retrieving vast volumes of scientific and technical literature. Research on recurrent neural network-based document clustering algorithms for use in a wide variety of specialized document query services is on the horizon.

References

1. T. Jo, “Semantic string operation for specializing AHC algorithm for text clustering,” Annals of Mathematics and Artificial Intelligence, vol. 88, no. 10, pp. 1083–1100, 2020.

2. L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A combination of objective functions and hybrid Krill herd algorithm for text document clustering analysis," Engineering Applications of Artificial Intelligence, vol. 73, pp. 111–125, 2018

3. C. Tang, H. Wang, Z. Wang, X. Zeng, H. Yan, and Y Xiao, "An improved OPTICS clustering algorithm for discovering clusters with uneven densities," Intelligent Data Analysis, vol. 25, no. 6, pp. 1453–1471, 2021.

4. Igor Škrjanc, Goran Andonovski, José Antonio Iglesias, María Paz Sesmero, Araceli Sanchis,Evolving Gaussian on-line clustering in social network analysis,Expert Systems with Applications,Volume 207,2022,117881.

5. N. D. Dat, V. N. Phu, V. T. N. Tran, V. T. N. Chau, and T. A. Nguyen, "STING algorithm used English sentiment classification in a parallel environment," International Journal of Pattern Recognition and Artificial Intelligence, vol. 31, no. 07, Article ID 1750021, 2017

6. Zurauskiene, J., and Yau, C. (2016). pcaReduce: Hierarchical Clustering of Single-Cell Transcriptional Profiles. BMC Bioinform 17, 140.

7. Bonner RE. On some clustering technique. IBM J Res Dev 1964;8(1):22–32.

8. Hennig C, Meila M, Murtagh F, Rocci R. Handbook of cluster analysis. Handbook of modern statistical methods, New York, USA: Chapman & Hall/CRC Press; 2015, p. 730.

9. Kleinberg J. An impossibility theorem for clustering. In: Advances in neural information processing systems, 15, MIT Press; 2003, p. 463–70,

10. Fisher DH. Knowledge acquisition via incremental conceptual clustering. Mach Learn 1987;2(2):139–72.

11. Arabie P, Hubert LJ, De Soete G. Classification and clustering. Singapore: World Scientific; 1996, http://dx.doi.org/10.1007/s003579900026.

12. Duda RO, Hart PE, Stork DG. Pattern classification. 2nd ed.. Ney York, USA: A Wiley-Interscience Publication. John Wiley & Sons; 2001.

13. Everitt BS, Landau S, Leese M. Cluster analysis. 4th ed.. London: Arnold; 2001.

14. Handl J, Knowles J, Kell DB. Computational cluster validation in postgenomic data analysis. Bioinformatics 2005;21 (15):3201–12.