Materials Today: Proceedings xxx (xxxx) xxx

Contents lists available at ScienceDirect

Materials Today: Proceedings

journal homepage: www.elsevier.com/locate/matpr

# Performance evaluation of clustering algorithms for varying cardinality and dimensionality of data sets

Shini Renjith [a,b,*], A. Sreekumar [a], M. Jathavedan [a]

[a] Department of Computer Applications, Cochin University of Science and Technology, Kochi, Kerala 682022, India
[b] Department of Computer Science and Engineering, Mar Baselios College of Engineering and Technology, Thiruvananthapuram 695015, India

ARTICLE INFO

*Article history:*
Received 8 December 2019
Received in revised form 3 January 2020
Accepted 5 January 2020
Available online xxxx

*Keywords:*
Clustering algorithms
Clustering quality
Clustering performance
Social media
Turnaround time

ABSTRACT

Clustering is the most widely used unsupervised machine learning technique, having extensive applications in statistical analysis. We have multiple clustering algorithms available in theory and many more implementations available in practice. A bunch of literatures can be found focusing on the quality of clustering algorithms using various internal and external evaluation techniques. The motivation behind this work is the scarcity of literatures dealing with performance of clustering algorithms in terms of turnaround time. This paper summarizes the experimental analysis conducted on the performance of multiple clustering algorithms based on cardinality and dimensionality. The analysis is performed in R, which is a free and open source programming language mainly used for statistical computing. This work evaluates nine key algorithms coming under partitioning, hierarchical, density-based and model-based clustering approaches using different social media data sets. We captured performance trends of these algorithms in terms of turnaround time by varying the cardinality and dimensionality parameters of the data sets. Based on our experiments, CLARA, CLARANS, and k-means algorithms demonstrate best performances with varying cardinality. It is also observed that changes in dimensionality do not impact hierarchical clustering approaches whereas there is a positive influence on the execution time for partitioning, density-based and model-based clustering approaches.

Selection and Peer-review under responsibility of the scientific committee of the First International Conference on Recent Advances in Materials and Manufacturing 2019.

## 1. Introduction

Data mining [1] is the process of extracting meaningful information from raw data through which underlying patterns and relationships are revealed. These revelations form useful knowledge that can be made use of various scientific, educational, and/or industrial scenarios. Based on the type of patterns to be processed, we can adopt appropriate data mining strategies which include, but not limited to classification, clustering, association, regression, etc.

Clustering is the machine learning technique used for creating logical groups of similar entities from a data set. The aim of clustering process is to create distinct groups of elements in such a way that the entities from the same group will have similar properties whereas entities from different groups have dissimilar properties. It is an unsupervised learning technique which is widely used for per-forming statistical analysis of data. Since the volume of data being processed is increasing on a daily basis, clustering is extensively applied in almost all industrial segments.

This work covers an empirical analysis of the performance of nine different clustering algorithms [2]. We captured the average processing time for each algorithm against varying number of records (cardinality) with constant number of attributes (dimensionality), and varying number of attributes with same number of records. The experiments were conducted using two distinct social media data sets.

Section 2 of this paper does a quick recap of various clustering algorithms that are evaluated through this empirical analysis. Sections 3 summarizes the related literature and Section 4 briefs on the research methodology adopted, infrastructure and tools leveraged, and details of data sets used for evaluation. Section 5 records the statistical observations from the experiments. Section 6 concludes the paper with details of our inferences and future steps.

* Corresponding author at: Department of Computer Applications, Cochin University of Science and Technology, Kochi, Kerala 682022, India.
*E-mail address:* shinirenjith@gmail.com (S. Renjith).

https://doi.org/10.1016/j.matpr.2020.01.110
2214-7853/
Selection and Peer-review under responsibility of the scientific committee of the First International Conference on Recent Advances in Materials and Manufacturing 2019.

## 2. Antecedents

### 2.1. Clustering

Clustering is the process of segregating a data set into different groups called clusters which contains similar entities in it. Based on the clustering model adopted, clustering algorithms can be generally classified into either partitioning, hierarchical, density-based, or model-based clustering. In this work we considered nine most common algorithms for evaluation–five partitioning algorithms, two hierarchical algorithms, one density-based algorithm, and one model-based algorithm.

### 2.2. k-means clustering

k-means clustering algorithm [3–6] organizes the entities in a given data set into k distinct clusters through an iterative process. The algorithm attempts to yield the minimum value for the total within cluster variation of the clusters being populated. The total within cluster variation can be computed as the sum of the squared error for all the entities in the data set. Mathematically it can be represented as (1).

$$Total\ WCV = \sum_{k=1}^{K} \sum_{E_i \in C_k} (E_i - \mu_k)^2 \tag{1}$$

where $K$ is the total number of clusters formed and $E_i$ is an element in cluster, $C_k$ having centroid, $\mu_k$.

### 2.3. k-medoids clustering

k-medoids algorithm [7,8] differs from k-means algorithm mainly in the way in which the cluster centroids are arrived at. Unlike k-means algorithm, k-medoids algorithm always select an entity from the data set as the centroid (called medoid as it indicates the median in Statistics) in such a way that the sum of dissimilarities with every other entities of the same cluster is minimum. The most common k-medoids algorithm is called Partitioning around Medoids (PAM). The sum of dissimilarities across all clusters can be computed as the sum of absolute error for all the entities in the data set. Mathematically it can be represented as (2).

$$Cumulated\ Sum\ of\ Dissimilarities = \sum_{k=1}^{K} \sum_{E_i \in C_k} \|E_i - \mu_k\| \tag{2}$$

where $K$ is the cluster count and $E_i$ is an element in cluster, $C_k$ having medoid, $\mu_k$.

### 2.4. Clustering large applications

Clustering large applications (CLARA) algorithm [7,9] is created as an extension to PAM with the intention of dealing with large data sets. The algorithm works by drawing samples from the data set and applying PAM on each sample to choose the corresponding set of medoids. The goodness of these medoids are computed against the full data set using an objective function (average dissimilarity between every entity in the data set and the medoid of the cluster it belongs to). This sampling and clustering process is repeated for a defined number of iterations and the clusters corresponding to the set of medoids which results in minimum value for objective function is identified. Mathematically the objective function can be represented as (3).

$$Objective\ Function(M,\ D) = \sum_{i=1}^{N} \frac{d(E_i,\ medoid(M,\ E_i))}{N} \tag{3}$$

where $M$ is the set of medoids in consideration, $D$ is the complete data set having $N$ number of entities in it, $E_i$ is an element in $D$, $medoid(M,\ E_i)$ is the medoid chosen from $M$ which is nearest to $E_i$, and $d(x,\ y)$ is the dissimilarity between $x$ and $y$.

### 2.5. Clustering large applications based upon randomized search

Clustering large applications based upon randomized search (CLARANS) [10] is another extension to PAM algorithm. While CLARA proceed with a fixed sample chosen at random, CLARANS proceeds with randomness included in every step of its processing by choosing a neighbor dynamically. It is like a graph search problem with every node representing a potential solution, which is a set of medoids. Two nodes are considered as neighbors when their corresponding set of medoids differ by only one entity. With the new neighbor selected, if the local optimum is found, the search is continued with the newly selected node else with the same node.

### 2.6. Fuzzy c-means clustering

Fuzzy C-means algorithm [11–14] allows every entity in the data set to be part of every cluster being formed, to a certain magnitude. The degree of belonging of an entity to a particular cluster is determined by the similarity of the entity with the centroid of the cluster. The more near to a particular centroid, the entity will have a higher belonging to the corresponding cluster. The cumulative belonging across all clusters for any entity at any time is kept as 1 or 100%. Mathematical representation of the same is shown in (4).

$$\sum_{k=1}^{K} M_k(E_i) = 1 \tag{4}$$

where $K$ is total number of clusters and $M_k(E_i)$ is the degree of membership of an element $E_i$ in cluster $C_k$.

### 2.7. Agglomerative hierarchical clustering

Agglomerative hierarchical clustering (AGNES) algorithm [8,15–17] is a bottom-up clustering approach. Starting from singleton clusters the algorithm the algorithm continuously merges the nearest clusters together till it reaches a single cluster. The major challenge of this algorithm is the lack of global distribution details at early stages of clustering, which results in irreversible clustering decisions based on local patterns.

### 2.8. Divisive hierarchical clustering

Divisive Hierarchical Clustering (DIANA) algorithm [8,17] adopts a top-down clustering approach. Starting with a single cluster containing full data set, the algorithm recursively proceeds till reaching singleton clusters. At each stage a flat clustering algorithm like k-means is used for slicing parent clusters.

### 2.9. Density-based spatial clustering of applications with noise

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [18,19] is the most extensively used density based clustering algorithm. The algorithm group entities in a data set in such a way that entities that are closely located are considered as a single cluster and entities that are present in low density areas are treated as noises or outliers. The main issue with DBSCAN algorithm is its sensitivity to the parameters like cluster radius, epsilon and minimum number of entities required in a cluster.

Please cite this article as: S. Renjith, A. Sreekumar and M. Jathavedan, Performance evaluation of clustering algorithms for varying cardinality and dimensionality of data sets, Materials Today: Proceedings, https://doi.org/10.1016/j.matpr.2020.01.110

### 2.10. Expectation-maximization clustering

The expectation-maximization (EM) clustering algorithm [20,21] is a model-based clustering approach where probability of cluster memberships for each entity is calculated based on probability distribution models. The goal of the algorithm is to maximize the cumulative probability for all elements in the data set. The EM clustering algorithm assumes the data set to be a subset of Gaussian distribution mixture.

## 3. Related works

In 2005, Xu et al. [22] published a survey on various clustering algorithms, but they did not attempt to cover the big data context. Shirkhorshidi et al. [23], in 2014 came up with a detailed review of big data clustering approaches. Other key contributions on analyzing big data clustering include the works from Sajana et al. [24], Ajin et al. [25], and Dave et al. [26]. These are good literature on the theoretical aspects of various clustering algorithms but lacks the experimental analysis.

One of the early attempts on empirical analysis of clustering techniques was performed in 1998 by Lau et al. [27] who compared two unsupervised neural network clustering algorithms for their performance in information retrieval from image databases. In 2002, Maulik et al. [28] evaluated the clustering performances of k-means clustering algorithm, single linkage clustering scheme, and a simulated annealing based technique using internal evaluation criteria of Davies-Bouldin index, Dunn's index, Calinski-Harabasz index, and index I. In 2003, Wei et al. [29] conducted experiments with CLARA, CLARANS, GAC-R3, and GAC-RAR to compare their performance. Again in 2003, Zhang [30] came up with k-harmonic means clustering algorithm and compared its performance against k-means and EM algorithms.

In 2005, Wang et al. [31] published comparison of DBSCAN and DBRS algorithms covering both theoretical and empirical aspects. In 2012, Poonam et al. [32] compared the performance of PAM, CLARA, CLARANS, and Fuzzy C-Means clustering algorithms for their outlier detection efficiency. Another experimental attempt was done by Fahad et al. [33] in 2014 by evaluating the clustering quality of five candidate clustering algorithms (DENCLUE, OptiGrid, Fuzzy C-Means, EM, and BIRCH) against ten different data sets. Again in 2014, Jung et al. [34] applied logistic regression analysis to compare clusters formed via EM and the k-means algorithms.

In addition to these generic experiments, there are a few attempts recorded which explicitly focuses on specific domains. In 2017, Bhatnagar et al. [35] performed a comparative evaluation of the performance of k-means clustering, hierarchical clustering, Fuzzy c-means clustering, Gaussian Mixture Modeling, and Self Organized Map clustering for the purpose of grouping manufacturing firms. In 2018, Renjith et al. [36] did empirical studies on clustering quality of various algorithms explicitly focusing on tourism data from social media channels.

## 4. Methodology

### 4.1. Approach

We have adopted a three stage approach in this research as explained below:

- *Determination of Optimal Cluster Count*. Partitioning and model-based approaches require to specify the optimal number of clusters as an input for clustering process. The optimal number of clusters is specific to the data set under consideration and highly subjective to the similarity measures being used. We have leveraged the R package NbClust [37], which computes 30 different indices to determine the optimal cluster count.
- *Clustering*. We have leveraged the most frequently used variations of clustering algorithms available in R for performing the analysis. We have performed multiple iterations of clustering with each of the data sets by varying cardinality and dimensionality parameters. We have evaluated the quality aspect of different clustering algorithms in our previous work [36] and here our key focus is on the turnaround time required for clustering algorithms.
- *Performance Evaluation*. We have captured the average execution time for each of the clustering algorithms under two scenarios in order to measure the performance of the algorithms in terms of turnaround time. In the first scenario, we attempted to detect underlying performance against changes in cardinality of the data sets. Second scenario aimed at detecting potential performance patterns for each of the algorithms by varying the dimensionality.

### 4.2. Tools used

R programming language [38,39], the free open source platform for statistical computing and data representation and RStudio [40], the integrated development environment for R are the main tools used in this experiment. The experiments are conducted on Intel Core i5-5200U, 2.20 GHz dual core x64-based processor with 8.00 GB RAM.

### 4.3. Data sets used

We considered two real-time data sets for the analysis and captured the results. Data set 1 contains user ratings from Google reviews collated by [36]. This data set contains anonymous reviews on 24 categories of attractions across Europe where user ratings are ranging from 1 to 5. Data sets 2 contains anonymous ratings from the Jester Online Joke Recommender System [41]. This data

**Table 1**
Summary of data sets used for analysis.

| Details | Data Set 1 | Data Set 2 |
|---|---|---|
| Data Set Description | User's average rating information on various types of attractions in Europe captured from a social media channel | Anonymous ratings from the Jester Online Joke Recommender System |
| Source | Google Destination Reviews | Jester Online Joke Recommender System |
| Cardinality | 5456 | 73,421 |
| Dimension | 24 | 100 |
| No. of Ratings | Around 1.3 lakhs | Around 4.1 million |
| Range | 1 to 5 | −10 to 10 |

Please cite this article as: S. Renjith, A. Sreekumar and M. Jathavedan, Performance evaluation of clustering algorithms for varying cardinality and dimensionality of data sets, Materials Today: Proceedings, https://doi.org/10.1016/j.matpr.2020.01.110

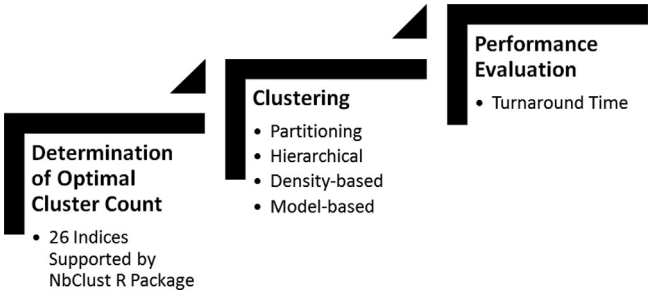


**Fig. 1.** Three stage research methodology adopted.

set contains around 4.1 million ratings ranging from −10.00 to +10.00 corresponding to 100 jokes by 73,421 users which are collected during April 1999 to May 2003. In addition, we extracted three subsets of data set 2 with reduced dimension (with 25, 50 and 75 attributes) for performing additional experiments with varying dimensions and fixed cardinality. A quick summary of the data sets in consideration is available in Table 1.

## 5. Empirical research

We have adopted a three stage approach in this research as depicted in Fig. 1.

### 5.1. Determination of optimal cluster count

We leveraged 26 different indices from the R package NbClust to decide the optimal count of clusters for each data set. The results corresponding to each data set are captured in Fig. 2. The optimal number of clusters are identified as 4 for data set 1 and 3 for data set 2.

### 5.2. Clustering

We have covered nine clustering algorithms as part of this analysis – five partitioning algorithms, two hierarchical algorithms, one density-based algorithm, and one model-based algorithm. Table 2 summarizes the details of the algorithms along with corre-

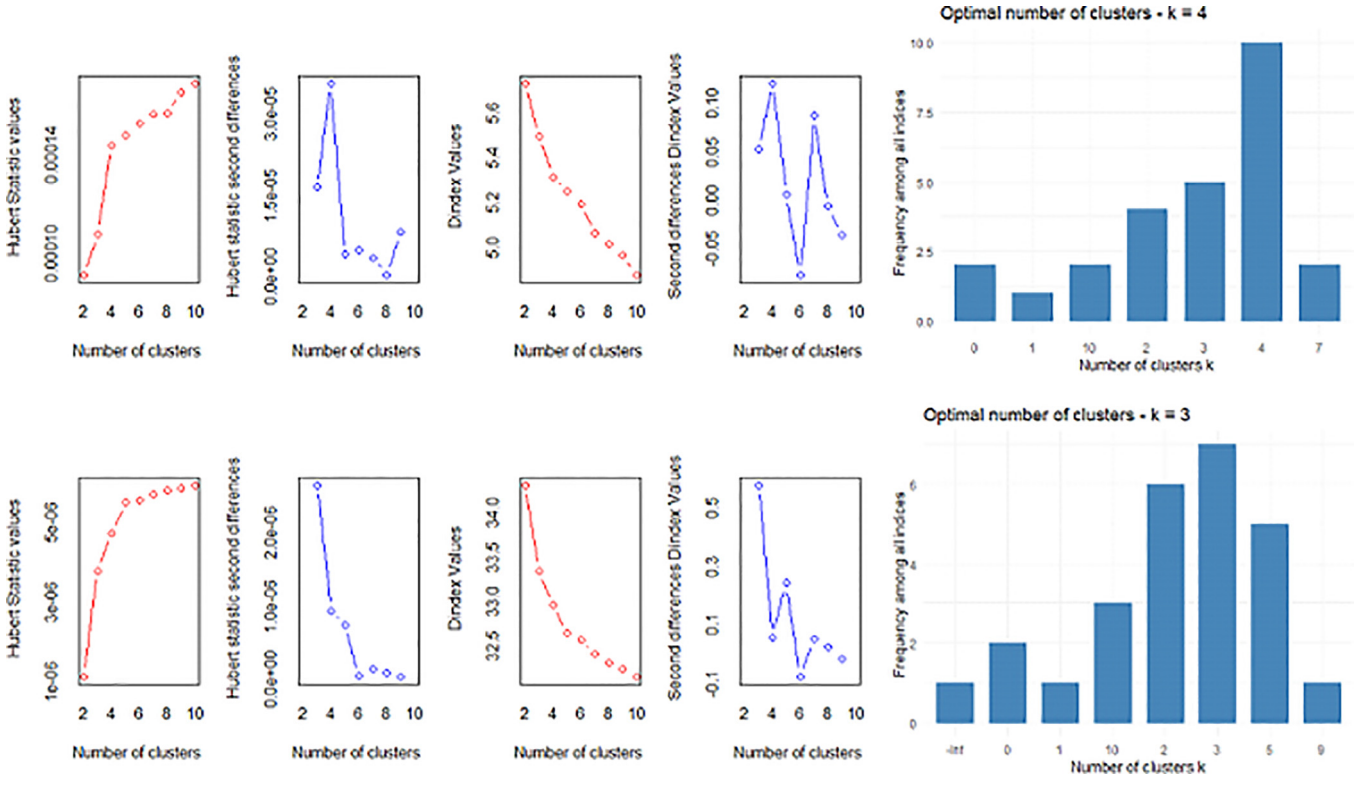


**Fig. 2.** Determination of optimal count of clusters for data sets used.

**Table 2**
Clustering algorithms used.

| Clustering Algorithm | Clustering Type | R function used to perform clustering | Applicable R package |
|---|---|---|---|
| K-Means Clustering | Partitioning | kmeans() | stats |
| k-medoids Clustering | Partitioning | pam() | cluster |
| Clustering Large Applications | Partitioning | clara() | cluster |
| Clustering Large Applications based upon Randomized Search | Partitioning | clara() | cluster |
| Fuzzy c-means Clustering | Partitioning | fcm() | ppclust |
| Agglomerative Hierarchical Clustering | Hierarchical | agnes() | cluster |
| Divisive Hierarchical Clustering | Hierarchical | diana() | cluster |
| Density-Based Spatial Clustering of Applications with Noise | Density based | dbscan() | fpc |
| Expectation-Maximization Clustering | Model based | hddc() | HDclassif [42] |

Please cite this article as: S. Renjith, A. Sreekumar and M. Jathavedan, Performance evaluation of clustering algorithms for varying cardinality and dimensionality of data sets, Materials Today: Proceedings, https://doi.org/10.1016/j.matpr.2020.01.110

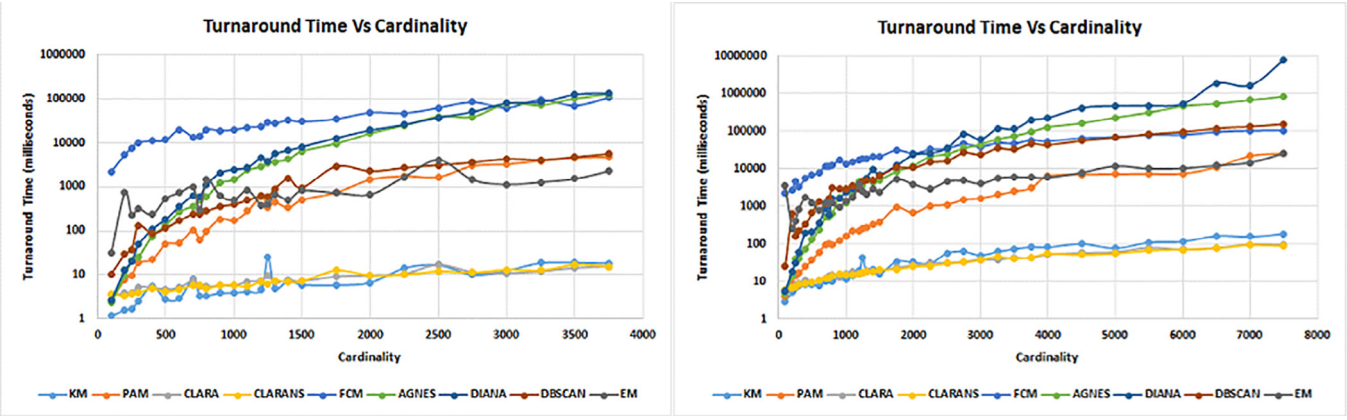


**Fig. 3.** Performance of clustering algorithms – Cardinality Vs Turnaround Time.

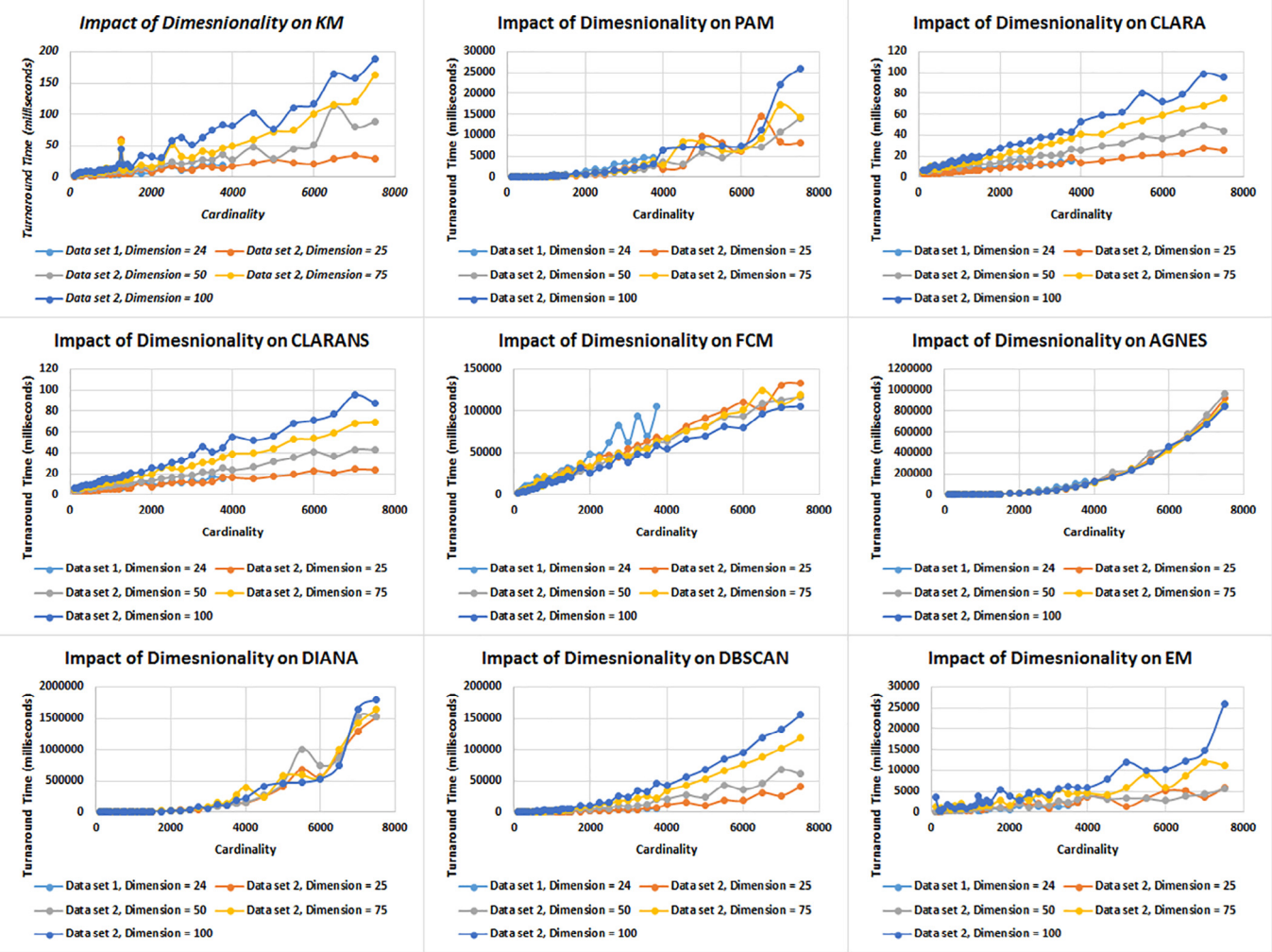


**Fig. 4.** Performance of clustering algorithms – Dimensionality Vs Turnaround Time.

sponding R function calls and package details leveraged in this evaluation.

### 5.3. Performance evaluation

We evaluated two types of performance impacts for clustering as part of this empirical study, i.e. impact on varying cardinality and impact of varying dimensionality.

- *Cardinality Vs Turnaround Time*. As part of this analysis, we compared all nine clustering algorithms for their performance by varying the cardinality of same data sets. Cardinality is varied by applying random sampling on the data sets. For data set 1, we have performed 27 iterations with cardinality ranging from 100 to 3750 and for data set 2, we have performed 35 iterations with cardinality ranging from 100 to 7500. During each iteration, we captured the mean turnaround time for the clustering algo-

Please cite this article as: S. Renjith, A. Sreekumar and M. Jathavedan, Performance evaluation of clustering algorithms for varying cardinality and dimensionality of data sets, Materials Today: Proceedings, https://doi.org/10.1016/j.matpr.2020.01.110

rithm at millisecond level using the microbenchmark package available in R. The best performances are observed with CLARA, CLARANS, and k-means algorithms. The observations are consistent across both the data sets considered. Fig. 3 depicts the plots of the observed turnaround time against cardinality for two data sets. Turnaround time is measured in milliseconds and represented on logarithmic scale in the plot to avoid skewness towards large values present in the observation set.

- *Dimensionality Vs Turnaround Time*. As part of this analysis, we compared all nine clustering algorithms individually for their performance by varying the dimensionality. Data set 1 has 24 attributes and data set 2 has 100 attributes. In addition, we formed three additional data sets by varying the number of attributes in data set 2. Fig. 4 depicts the variation in turnaround time for all the five data sets in consideration with different dimensions. It can be observed from the plots that hierarchical clustering approaches resulting in the same performance trends in spite of any change in dimensionality of the data sets being processed.

## 6. Conclusion

Large volume of social media data is getting generated vigorously, which has the potential to function as a key input for various analytics and data science activities across industries. In order to deal with the huge volume of data, it is required to adopt an efficient processing mechanism which ensures only relevant data is being processed. Various clustering algorithms can be considered as a possible solution to address this challenge. There are two important factors to be considered while selecting the appropriate clustering algorithms. The most important aspect is the clustering quality which has to be chosen based on the evaluation of data set in consideration [36]. At the same time, we need to ensure the algorithms are efficient in terms of their performance. In this work we evaluated nine most frequently used algorithms for their performance in terms of turnaround time. We used two real time data sets reflecting social media activities and performed an empirical study by varying the cardinality and dimensionality of the data sets.

In the context of this empirical analysis, we could observe better performances in terms of overall time taken from CLARA, CLARANS, and k-means algorithms. Further we observed the highest turnaround time with Fuzzy c-means algorithm. Hierarchical clustering algorithms (AGNES and DIANA) and DBSCAN demonstrated an increasing trend similar to that of power functions for the overall time taken with increase in cardinality. Other algorithms demonstrated a linearly increasing trend with increase in cardinality. While other algorithms demonstrated a positive influence on the execution time for the change in dimensionality, both the hierarchical clustering algorithms (AGNES and DIANA) demonstrated neutral impact.

We recommend considering these observations along with clustering quality parameters while performing selection of clustering techniques in any real time scenario. Further, the simple and repeatable process defined as part of this experiments can be adopted by researchers for evaluating any new algorithms and/or data sets.

## CRediT authorship contribution statement

**Shini Renjith:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization. **A. Sreekumar:** Supervision, Writing - review & editing. **M. Jathavedan:** Supervision, Writing - review & editing.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] M. Kantardzic, Data Mining: Concepts, Models, Methods, and Algorithms, John Wiley & Sons, 2011.
[2] V. Estivill-Castro, Why so many clustering algorithms, ACM SIGKDD Explor. Newslett. 4 (1) (2002) 65–75, https://doi.org/10.1145/568574.568575.
[3] J. Hartigan, M. Wong, Algorithm AS 136: a K-means clustering algorithm, J. Roy. Stat. Soc.: Ser. C (Appl. Stat.) 28 (1) (1979) 100–108, https://doi.org/10.2307/2346830.
[4] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, vol. 1, no. 14 (1967) 281–297.
[5] S. Lloyd, Least squares quantization in PCM, IEEE Trans. Inf. Theory 28 (2) (1982) 129–137, https://doi.org/10.1109/TIT.1982.1056489.
[6] E. Forgey, Cluster analysis of multivariate data: efficiency vs interpretability of classification, Biometrics 21 (3) (1965) 768–769.
[7] L. Kaufman, P. Rousseeuw, Clustering by Means of Medoids, Faculty of Mathematics and Informatics, Delft, North-Holland, 1987.
[8] L. Kaufman, P. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, 2009, doi: 10.1002/9780470316801.
[9] H. Park, C. Jun, A simple and fast algorithm for k-medoids clustering, Expert Syst. Appl. 36 (2) (2009) 3336–3341, https://doi.org/10.1016/J.ESWA.2008.01.039.
[10] R. Ng, J. Han, CLARANS: a method for clustering objects for spatial data mining, IEEE Trans. Knowl. Data Eng. 14 (5) (2002) 1003–1016, https://doi.org/10.1109/TKDE.2002.1033770.
[11] J. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, J. Cybernet. 3 (3) (1973) 32–57, https://doi.org/10.1080/01969727308546046.
[12] J. Bezdek, Cluster validity with fuzzy sets, J. Cybernet. 3 (3) (1973) 58–73, https://doi.org/10.1080/01969727308546047.
[13] J. Bezdek, R. Ehrlich, W. Full, FCM: the fuzzy c-means clustering algorithm, Comput. Geosci. 10 (2–3) (1984) 191–203, https://doi.org/10.1016/0098-3004(84)90020-7.
[14] J. Bezdek, Corrections for "FCM: the fuzzy c-means clustering algorithm", Comput. Geosci. 11 (5) (1985) 660, https://doi.org/10.1016/0098-3004(85)90094-9.
[15] A. Lukasová, Hierarchical agglomerative clustering procedure, Pattern Recogn. 11 (5–6) (1979) 365–381, https://doi.org/10.1016/0031-3203(79)90049-9.
[16] M. Zepeda-Mendoza, O. Resendis-Antonio, Hierarchical agglomerative clustering, Encyclopedia Sys. Biol. 886–887 (2013), https://doi.org/10.1007/978-1-4419-9863-7_1371.
[17] M. Roux, A comparative study of divisive and agglomerative hierarchical clustering algorithms, J. Classif. 35 (2) (2018) 345–366, https://doi.org/10.1007/S00357-018-9259-9.
[18] M. Ester, H. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: The Second International Conference on Knowledge Discovery and Data Mining (KDD'96), AAAI Press, Portland, OR, USA, 1996, pp. 226–231.
[19] E. Schubert, J. Sander, M. Ester, H. Kriegel, X. Xu, DBSCAN revisited, revisited: why and how you should (still) use DBSCAN, ACM Trans. Database Syst. 42 (3) (2017) 1–21, https://doi.org/10.1145/3068335.
[20] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Stat. Soc.: Ser. B (Methodol.) 39 (1) (1977) 1–38.
[21] C. Bouveyron, S. Girard, C. Schmid, High-dimensional data clustering, Comput. Stat. Data Anal. 52 (1) (2007) 502–519, https://doi.org/10.1016/J.CSDA.2007.02.009.
[22] R. Xu, D. WunschII, Survey of clustering algorithms, IEEE Trans. Neural Networks 16 (3) (2005) 645–678, https://doi.org/10.1109/TNN.2005.845141.
[23] A. Shirkhorshidi, S. Aghabozorgi, T. Wah, T. Herawan, Big data clustering: a review, in: The 14th International Conference on Computational Science and Its Applications – ICCSA 2014, Springer International Publishing, Guimaraes, Portugal, 2014, pp. 707–720, doi: 10.1007/978-3-319-09156-3_49.
[24] T. Sajana, C. Sheela Rani, K. Narayana, A survey on clustering techniques for big data mining, Indian J. Sci. Technol. 9 (3) (2016) 1–12, https://doi.org/10.17485/IJST/2016/V9I3/75971.
[25] V. Ajin, L. Kumar, Big data and clustering algorithms, in: 2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS), IEEE Press, Bangalore, India, 2016, pp. 101–106, doi: 10.1109/RAINS.2016.7764405.
[26] M. Dave, H. Gianey, Different clustering algorithms for big data analytics: a review, in: 2016 International Conference System Modeling & Advancement in Research Trends (SMART), IEEE Press, Moradabad, India, 2016, pp. 328–333, doi: 10.1109/SYSMART.2016.7894544.
[27] T. Lau, I. King, Performance analysis of clustering algorithms for information retrieval in image databases, in: 1998 IEEE International Joint Conference on Neural Networks Proceedings, IEEE World Congress on Computational

Please cite this article as: S. Renjith, A. Sreekumar and M. Jathavedan, Performance evaluation of clustering algorithms for varying cardinality and dimensionality of data sets, Materials Today: Proceedings, https://doi.org/10.1016/j.matpr.2020.01.110

Intelligence (Cat. No.98CH36227), IEEE Press, Anchorage, AK, USA, pp. 932–937, doi: 10.1109/IJCNN.1998.685895.

[28] U. Maulik, S. Bandyopadhyay, Performance evaluation of some clustering algorithms and validity indices, IEEE Trans. Pattern Anal. Mach. Intell. 24 (12) (2002) 1650–1654.

[29] C. Wei, Y. Lee, C. Hsu, Empirical comparison of fast partitioning-based clustering algorithms for large data sets, Expert Syst. Appl. 24 (4) (2003) 351–363, https://doi.org/10.1016/S0957-4174(02)00185-9.

[30] B. Zhang, Comparison of the Performance of Center-Based Clustering Algorithms, in: Advances in Knowledge Discovery and Data Mining, PAKDD 2003, Lecture Notes in Computer Science, Springer, Seoul, Republic of Korea, 2003, pp. 63–74, doi: 10.1007/3-540-36175-8_7.

[31] X. Wang, H. Hamilton, A comparative study of two density-based spatial clustering algorithms for very large datasets, in: Advances in Artificial Intelligence, AI 2005, Lecture Notes in Computer Science, Springer, Victoria, BC, Canada, 2005, pp. 120–132, doi: 10.1007/11424918_14.

[32] Poonam, M. Dutta, Performance analysis of clustering methods for outlier detection, in: 2012 Second International Conference on Advanced Computing & Communication Technologies (ACCT 2012), IEEE Press, Rohtak, India, 2012, pp. 89–95, doi: 10.1109/ACCT.2012.84.

[33] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Zomaya, S. Foufou, A. Bouras, A survey of clustering algorithms for big data: taxonomy and empirical analysis, IEEE Trans. Emerging Top. Comput. 2 (3) (2014) 267–279, https://doi.org/10.1109/TETC.2014.2330519.

[34] Y. Jung, M. Kang, J. Heo, Clustering performance comparison using k-means and expectation maximization algorithms, Biotechnol. Biotechnol. Equip. 28 (2) (2014) S44–S48, https://doi.org/10.1080/13102818.2014.949045.

[35] V. Bhatnagar, R. Majhi, P. Jena, Comparative performance evaluation of clustering algorithms for grouping manufacturing firms, Arabian J. Sci. Eng. 43 (8) (2017) 4071–4083, https://doi.org/10.1007/S13369-017-2788-4.

[36] S. Renjith, A. Sreekumar, M. Jathavedan, Evaluation of partitioning clustering algorithms for processing social media data in tourism domain, in: 2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS), IEEE Press, Thiruvananthapuram, India, 2018, pp. 127–131, https://doi.org/10.1109/RAICS.2018.8635080.

[37] M. Charrad, N. Ghazzali, V. Boiteau, A. Niknafs, NbClust: an R package for determining the relevant number of clusters in a data set, J. Stat. Softw. 61 (6) (2014) 1–36, https://doi.org/10.18637/JSS.V061.I06.

[38] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2009.

[39] L. Tierney, The R statistical computing environment, Lect. Notes Statistics 435–447 (2012), https://doi.org/10.1007/978-1-4614-3520-4_41.

[40] J. Racine, RStudio: a platform-independent IDE for R and sweave, J. Appl. Economet. 27 (1) (2011) 167–172, https://doi.org/10.1002/JAE.1278.

[41] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Eigentaste: a constant time collaborative filtering algorithm, Inf. Retrieval 4 (2) (2001) 133–151, https://doi.org/10.1023/A:1011419012209.

[42] L. Bergé, C. Bouveyron, S. Girard, HDclassif: an R package for model-based clustering and discriminant analysis of high-dimensional data, J. Stat. Softw. 46 (6) (2012) 1–29, https://doi.org/10.18637/JSS.V046.I06.

Please cite this article as: S. Renjith, A. Sreekumar and M. Jathavedan, Performance evaluation of clustering algorithms for varying cardinality and dimensionality of data sets, Materials Today: Proceedings, https://doi.org/10.1016/j.matpr.2020.01.110