ICIC Express Letters
Volume **17**, Number **12**, December **2023**
 ISSN 1881-803X
pp. **1405–1415**

DOCUMENTS CLUSTERING USING SUBSPACE CLUSTERING ALGORITHM

Mohamed Shenify[1] and Fokrul Alom Mazarbhuiya[2,*]

[1]College of Computer Science and IT
Albaha University
Albaha 65799, Saudi Arabia
maalshenify@bu.edu.sa

[2]School of Fundamental and Applied Sciences
Assam Don Bosco University
Tapesia Garden, Kamarkuchi, Sonapur, Guwahati, Assam 782402, India
*Corresponding author: fokrul.mazarbhuiya@ydbuniversity.ac.in

Received February 2023; accepted April 2023

Abstract. *Clustering is the process of assembling abstract objects that share alike characteristics. Clustering has been commonly used in several arenas such as market research, pattern recognition, data analysis, and image processing. Document clustering is also called text clustering, an extension to traditional clustering used to analyze textual documents. It has been widely used in automatic topic extraction, and filtering, etc. Off late, it has also been used in Natural Language Processing (NLP) as a tool for the usual division of text documents into dissimilar categories so that documents with the identical category will have high resemblance to those within different categories. Document clustering has been effectively used in several fields, such as mining web data, web searching, information retrieval, and Topological Data Analysis (TDA). Lately, some studies have used document clustering to understand the social behavior of social media users based on their posts. In this study, we recommend a clustering method that is used to find clusters from high-dimensional data using subspace clustering. The algorithm's efficacy is verified by experimental studies conducted on a real-life data from UCI Machine Repository.*
Keywords: Subspace clustering, Text documents, Dense cells, Density-based clustering, High-dimensional data

1. **Introduction.** Clustering [1] is an unsupervised approach to categorizing patterns into groups, and intends to classify the data into similar groups. In the machine learning and database communities, clustering is frequently employed. Its usage is increasing rapidly, such as fault finding, anomaly or outlier detection, and hacker identification [2-7]. Accordingly, it was applied in text mining, social media data or any other e-publications analysis. Most of the contents in social media or e-publications are either semi-structured or unstructured and are available in electronic form, and they require efficient methodology for analysis. An efficient document clustering algorithm can be employed for these purposes.

Document clustering [8] may be a useful tool for the analysis of electronic text data. With an unlabeled document collection, document clustering can be useful to classify the collection based on certain conditions. Initially, information retrieval activities used document clustering. However, most recently, social media data analysis has attracted the attention of researchers. The authors of [8] did a comparison study of various document clustering methodologies.

Text datasets are growing at very high speeds owing to the growing of volume of information in the electronic form, such as e-publications, email, and World Wide Web.

DOI: 10.24507/icicel.17.12.1405

Until now, researchers engrossed on structured data. However, text datasets are either unstructured or semi-structured; thus, mining such datasets can be the most challenging task. In [9], the authors analyzed some well-known swarm intelligence algorithms with k-means algorithm for the clustering of text document. Another important aspect of text datasets is the dimension of the datasets. Text mining problem and its applications in social media were extensively addressed in [10,11]. In most of the aforesaid works, the underlying datasets are either semi-structured or un-structured. In such datasets, the trivial clustering method does not work. Also, as the dataset size increases, the dimension of the dataset also increases proportionately. Therefore, finding clusters from high-dimensional text document data can be an interesting research area.

One approach to tackling this issue is to convert the dataset into a high-dimensional dataset, where a vector in a high-dimensional data [12] represents a document, which can be expressed with respect to frequencies of the rest of the terms within that document. Then, the clusters of any projections of space [12] can be extracted. This will give us clusters in a subspace of high-dimensional space, which can be used to distinguish between dense and sparse regions in the lower-dimensional realm.

We mentioned that a document is a vector of the leftover term's frequencies inside it and is represented by a point in a high-dimensional space. As we know that the complexity of any clustering algorithm increases proportionately with an increase in space dimensions, the clustering high-dimensional data is a bit challenging. One way to address this problem is to allow the users to specify the subspace for clustering [13], which would reduce the dimension of clusters. However, user identification of the subspace is not free from errors. Another method to tackle this problem is the use of Principal Component Analysis (PCA) or Kerhunen-Loeve (K-L) transformation [14,15]. Although the methods successfully reduced the dimensions, the new dimensions could not be interpreted simply. In addition, these techniques may not be useful for the detection of clusters inside different subspaces of the same space.

In this work, an algorithm is proposed which addresses the aforesaid issues efficiently. Our approach is a density-based approach that spontaneously discovers subspaces with high-density. This method is discussed briefly below.

Here, we are interested in identifying subspace from a high-dimensional dataset for improved clustering of data points compared to the original dataset. We employ the density-based algorithm, where a cluster would be a region of high-density. We partition the space into a couple of cells and calculate the amount of data points inside the cell. This could be done by splitting each dimension into equal, identically spaced intervals. Obviously, every cell will have a constant volume, so the total points belonging to a cell would be utilized to estimate the cell density. After obtaining the appropriate subspaces, clusters are discovered in the corresponding subspace. The valley of the density function separates the data points. The clusters can be considered as the union of connected high-density cells [12] inside a subspace. To simplify their explanations, we restrict the clusters to axis-aligned hyper-rectangles [12].

Each cell in a subspace of dimension k is the intersection of inequalities, as it is the intersection of $2k$-axis-aligned half-spaces characterized by k, 1-dimensional intervals. Accordingly, a cluster is described with Disjunctive Normal Form (DNF) expressions, as it is the union of the above-mentioned cells. By using a cover with the least amount of maximal, probably overlapping rectangles and then expressing the cluster as a union of such rectangles, a compact description of the cluster is obtained.

The algorithm discussed produces cluster descriptions such as DNF expression. It also tries to produce minimal descriptions for the simplicity of comprehension. The rest of this article is organized as follows. The most recent advancements in this area are covered in Section 2. The problem statement and preliminaries are presented in Section 3. Section 4 presents the proposed algorithm along with a flowchart. In Section 5, the complexity

analysis is presented. Section 6 presents the experimental analysis. Finally, in Section 7, we conclude the article with conclusions and directions for further research.

2. **Related Works.** Text in electronic form has become a vital part of day-to-day life as the volume is growing rapidly. Nowadays, many people have started using soft materials instead of hard materials, as most of the literature is published on different digital platforms and can be downloaded at any time from any place. Therefore, the amount of data piling up in various digital platforms is rapidly increasing. The difficulty is to mine such an enormous amount of data and draw inferences. This can be achieved using various data mining techniques [1]. Data clustering [1] is one such important data mining approaches applied in many fields. Mazarbhuiya et al. [2] proposed an agglomerative hierarchical algorithm for the anomaly detection in network data. In [3], the authors presented a hybrid clustering approach for the analysis of network data. In [4], the authors proposed a modified density-based algorithm for the analysis of time-series data. A clustering-based approach for the analysis of real-time data is proposed by Habeeb et al. [5]. Similar works were reported in [6,7].

In [10], an extensive study was conducted on various text mining on the social web. Using an optimized k-means algorithm [16], an algorithm for clustering social media data was proposed. In [17], the authors proposed a method that increases the search engines capacity for handling big data searching efficiently. In [18], the author suggested a technique for clustering high-dimensional data. In [19], a different method was presented for finding high-density clusters in a projection of high-dimensional data. In [12], the authors proposed an algorithm called CLIQUE, which automatically identifies high-density clusters in a projection. A new Sparse Subspace Clustering (SSC)-based community detection method for social network members was proposed in [20]. In [21], the authors presented a subspace clustering method that uses orderly relationships as a constraint to learn an orderly representation. The authors of [22] proposed an approach to handle the resulting optimization problem based on an augmented Lagrangian multiplier with alternating direction minimization.

In [23], the authors put forward a technique for the automatic discovery of a circle of friends on mobile data using combined clustering techniques. In [24], the authors presented an algorithm called SUBSCALE for subspace clustering with minimal cost. In [25], the authors proposed a method that can discover clusters in various subspaces in a single pass over a data stream. In [26], the authors suggested a subspace clustering framework that measures the local likelihood of samples in the same subspace and finds the global likelihood sample patterns for the manifold data types, thus broadly finding the heterogeneity of samples.

In [27], authors discussed about different existing metaheuristic machine learning algorithms that have already been used for several exciting research problems of social networks and big data. In [28], the authors demonstrated a neural network analysis of social network data which can be helpful for conducting interviews. In [29], the authors proposed a big data based technology for the classification of social media accounts. In [30], the authors studied data mining, recommender system, pattern recognition, etc., and suggested that pattern recognition can be applied as a classification tool for mining and classifying patterns. In [31], the authors proposed a machine learning approach to estimate whether and what magnitude well-known clustering algorithms are appropriate for the patterns recognition in multi-dimensional communication data. In [32], the authors proposed an approach to describe behavior of users by their association with the enterprise social media. In [33], a clustering-based manifold collaborative approach was used within social organizations to understand varied supervision in social organizations and increase the consequence of the concern pattern recognition.

3. **Problem Statement and Preliminaries.** We briefly go through the terminologies, definitions, and notations utilized in this method in this part.

3.1. **Definitions and notations associated to documents clustering.** The documents in the soft form were input for the mining process. Clustering such documents would help us in many ways, such as information retrieval, topic extraction, and document organization, in the analysis data of different digital platforms. As most of the contents of different digital platforms are not structured, they need to be preprocessed.

Any document d is a finite list consisting of the elements (w, n), where w is a keyword and n is the number of times w has occurred in d. If W is the collection of distinct keywords occurring in the documents such that $|W| = m$. Additionally, $W = \{w_1, w_2, w_3, \ldots, w_m\}$ appeared in some order. Any document d is written as $(n_1, n_2, n_3, \ldots, n_m)$, where n_i is the number of times w_i has occurred in d. Also, $n_i = 0$ for d if w_i is absent in d.

3.2. **Definitions and notations associated to subspace clustering.** Here, we are interested in clustering documents that are contents of text available in any digital platform. Each document is characterized by a vector in an n-dimensional space, that is, an n-dimensional point. As we employ a density-based methodology, a cluster would be a region with a higher point density.

For the computation of density, the cells are created from the data space by partitioning it, and the number of points within the cells is computed. This is done by dividing all dimensions into equal numbers of identical length intervals. Therefore, points within a cell can be used to estimate the cell density.

After finding a suitable subspace, the job is to discover clusters in the corresponding projection. The valleys of the density functions separate the data points. The unions of connected high-density cells in a subspace are clusters. For simplification of the descriptions of clusters, we limit them as hyper rectangles that are axis-aligned.

Every cell in a subspace of dimension n is expressed as a conjunction of inequalities, as they are the intersection of $2n$ axis aligned half-spaces characterized by n, 1-dimensinal intervals. As a cluster is the union of aforesaid cells, it may be expressed as a Disjunctive Normal Form expression. A compact description can be found by placing a cover over a cluster with the least amount of maximal, possibly overlapping rectangles, and expressing the cluster as a union of such rectangles. Below, we discuss the terminology related to subspace clustering.

Let $R = \{R_1, R_2, \ldots, R_d\}$ be a collection of bounded, totally ordered domains, and $S = R_1 \times R_2 \times \cdots \times R_d$ is a d-dimensional space where $R_1, R_2, \ldots, R_d$ are dimensions of S.

The input consists of a set of d-dimensional points $D = \{d_1, d_2, \ldots, d_m\}$, where each d_i is a document represented by a d-dimensional vector, $d_i = \{d_{i1}, d_{i2}, \ldots, d_{id}\}$, and the jth component of d_i is taken from domain R_j. We divide S into non-overlapping cells by dividing each dimension into t amounts of the same length intervals. Each cell $c = \{c_1, c_2, \ldots, c_d\}$ is the intersection of intervals one from each dimension, where $c_i = [l_i, r_i)$. We say that a document $d = \{d_1, d_2, \ldots, d_d\}$ is in cell c if $l_i \leq d_i \leq r_i$.

Density of cell c is the fraction of total documents present in it. Similarly, c is said to be dense if its density is more than or equal to a predetermined threshold θ (input parameter).

Similarly, we define the cells in each subspace of the given space of dimension d. We express a projection of dataset D as $A_{t1} \times A_{t2} \times \cdots \times A_{tk}$, $k < d$, $t_i < t_j$ if $i < j$. A cell in a projection is the intersection of intervals one from each k-dimension.

Any cluster can be considered as a maximal set of connected dense cells in k-dimensions. Two cells of dimension k say c_1 and c_2 are said to be connected if there is a common face between or there is another cell c_3 that is connected to both c_1 and c_2. The cells $c_1 = \{r_{t1}, r_{t2}, \ldots, r_{tk}\}$ and $c_2 = \{r'_{t1}, r'_{t2}, \ldots, r'_{tk}\}$ have a common face if there exist $k - 1$

dimensions such as $A_{t1} \times A_{t2} \times \cdots \times A_{tk-1}$, such that $r_{tj} = r'_{tj}$ and kth intervals will have a common side.

A k-dimensional region is an axis-aligned rectangular set of dimensions k. Here, our interest lies in those particular regions that could be represented as the union of cells and also expressed as a DNF expression on intervals of the domain A_i.

A region R is said to be contained in C (cluster) if $R \cap C = R$, and R is called maximal if C contains no proper subset of R. The cluster's minimal description is considered as a non-redundant cover with a maximal region.

4. **Proposed Algorithm.** The clustering of documents consists of the following steps:

1) Conversion of datasets into n-dimensional space;
2) Identification of subspaces containing clusters;
3) Finding of clusters;
4) Generation of minimal description of clusters.

At the beginning of the process, each document is converted into a vector with components as the frequencies of keywords. We assume that the keywords are arranged in a predefined order. If there are n keywords, then each document will be represented by n-dimensional vectors.

The method of finding subspaces is quite similar to the A priori [34,35] used for mining frequent itemsets. Thus, the algorithm for subspace identification is a level-wise process similar to that in [34,35]. First, it determines all the dense cells of dimension 1 by scanning the data. After determining the dense cells of dimension $(k-1)$, the candidate dense cells of dimension k are determined by the method discussed below. A dataset scan is made to find those dense candidate cells using a threshold θ [the definition of dense cell is given in Section 3]. The algorithm stops when a particular level is found to be empty or no more candidate is generated.

The candidate generation procedure takes as an input the set of all $(k-1)$-dimensional dense cells, D_{k-1}, and returns the candidate k-dimensional cells. Assuming $<$ represents the lexicographic ordering of the dimensions, we join D_{k-1} using conditions that first $(k-2)$-dimensions of the cells must be the same. If $c \cdot a_i$ is the ith dimension of cells c and $c \cdot [l_i, h_i)$, then the corresponding interval of c in the dimension. The pseudo-code is as follows:

Gen-candidate ($D_{k-1}c_1$, $D_{k-1}c_2$)
insert into C_k
select $c_1 \cdot [l_1, h_1), c_1 \cdot [l_2, h_2), \ldots, c_1 \cdot [l_{k-1}, h_{k-1}), c_2 \cdot [l_{k-1}, h_{k-1})$ from two $(k-1)$-dimensional cells $D_{k-1}c_1$, $D_{k-1}c_2$
where $c_1 \cdot a_i = c_2 \cdot a_i$, $c_1 \cdot l_i = c_2 \cdot l_i$, $c_1 \cdot h_i = c_2 \cdot h_i$; $i = 1, 2, \ldots, k-2$, and $c_1 \cdot a_{k-1} < c_2 \cdot a_{k-1}$

After candidate generation, it is required to make the algorithm faster. One way to minimize the number of passes is by pruning uninteresting candidates.

To find interesting subspaces and hence dense cells, the minimal description length principle [39] can be used. The basic idea of the minimal description length principle is to encode the input data under a given model and select the encoding that minimizes the code length. We assume that the given subspaces are S_1, S_2, and S_n. The pruning method first groups the dense cells that belong to the same subspace. Then, for each subspace, the data fraction that is covered by dense cells is computed. $xS_j = \sum_{c_i \in S_j} \text{count}(c_i)$, where $\text{count}(c_i)$ is the number of points that belongs to c_i, is the coverage of subspace S_j. The subspaces with maximum coverage were selected, and rests would be pruned.

After finding the subspaces, it is necessary to identify clusters in the subspaces. The algorithm for discovering clusters is similar to that discussed in [12], which is known as CLIQUE. The input to the algorithm is the set of dense cells D from the same space S of dimension k. The output of this is the partition of D into $D^1, D^2, \ldots, D^q$ such that each

cell of D_i is connected and no two cells $c_i \in D_i$, $c_j \in D_j$ with $i \neq j$ are connected. Each of these partitions is a cluster.

The problem of finding clusters is equivalent to the finding of connected components in a graph. In this notion, the dense cells are considered as nodes, and there is an edge between the nodes if the corresponding dense cells have common faces. Using the edge connectivity of the connected component and thus connected dense cells [the connectedness of cells is defined in Section 3] can be found. One connected component represents a cluster, as all the dense cells belonging to it are connected. Equivalently, the dense cells belonging to two different connected components are not connected, and hence they cannot belong to the same cluster. In fact, they were in two different clusters.

A depth-first-search algorithm [37] can be applied to find the connected component in a graph. Starting with a cell $c \in D$, and assigning it as a first cluster member, we try to find all the cells connected to it. Then, if there are still unvisited cells in D, we find one and repeat the process. The pseudo-code of the algorithm is written as follows:
input: starting $c = \{[l_i, h_i); i = 1, 2, \ldots, k\}$ cluster number n
  dfs(c, n)
$c \cdot num = n$
for $(i = 1; i < k; i++)$
$c^l == \{[l_1, h_1), \ldots, [l_i^l, h_i^l), \ldots, [l_k, h_k)\}$ // examine the left neighbor of c in dimension a_j
if (c^l is dense) and ($c^l \cdot num$ is undefined)
    dfs(c^l, n)
$c^r == \{[l_1, h_1), \ldots, [l_i^r, h_i^r), \ldots, [l_k, h_k)\}$ // examine the left neighbor of c in dimension a_j
if (c^r is dense) and ($c^r \cdot num$ is undefined)
    dfs(c^r, n)
end

As the clusters are the maximal set of cells in any subspace of the original space that are connected, they can be computed using the connectedness property of the cells [the connectedness of cells is defined in Section 3]. Taking the union of such connected cells, axis-parallel k-dimensional regions are computed, which are then expressed as DNF expression on intervals. Finally, the minimal description of the clusters was computed. The input to this step is the clusters that are in the form of a set of disjoint-connected cells of dimension k belonging to the same subspace. Our aim was to produce a concise description of it. For this purpose, the cells of the clusters are covered by the smallest number of regions. To do so, we followed two steps. First, we covered the cluster with a set of maximal regions. After this, we remove the redundant regions, which will give us a minimal cover. The method supplies the set of clusters S in a k-dimensional space, where each cluster is expressed by DNF. The flowchart of the algorithm is given in Figure 1.

5. **Complexity Analysis.** The running cost for converting a document to a vector is $O(n)$, where n is the total number of keywords. Therefore, the cost of converting the text dataset to n-dimensional space is $O(m \cdot n)$, where m is the total number of documents in the dataset. If a cell is dense, its projections in a subset of k-dimension are also dense. Obviously, there is $O(2^k)$ different projections. The algorithm has to make k pass over the dataset. The time complexity of identification of subspaces is $O(2^k + m \cdot k)$. For finding clusters, the algorithm needs to check $2k$ neighbors of each dense cell to find connected cells. If total number of dense cell in the subspace is p ($p \leq m$), the running time is $O(2p \cdot k) = O(p \cdot k)$. For, finding minimal cover descriptions, we need to cover the clusters with maximal regions (rectangles), and then discard the redundant regions which require the computational cost of $O(2c^2) = O(c^2)$, where c is the number of clusters covered by R. Therefore, the worst-case time complexity of the algorithm is $O(2^n + m \cdot k + p \cdot k + c^2) = O(2^n)$.

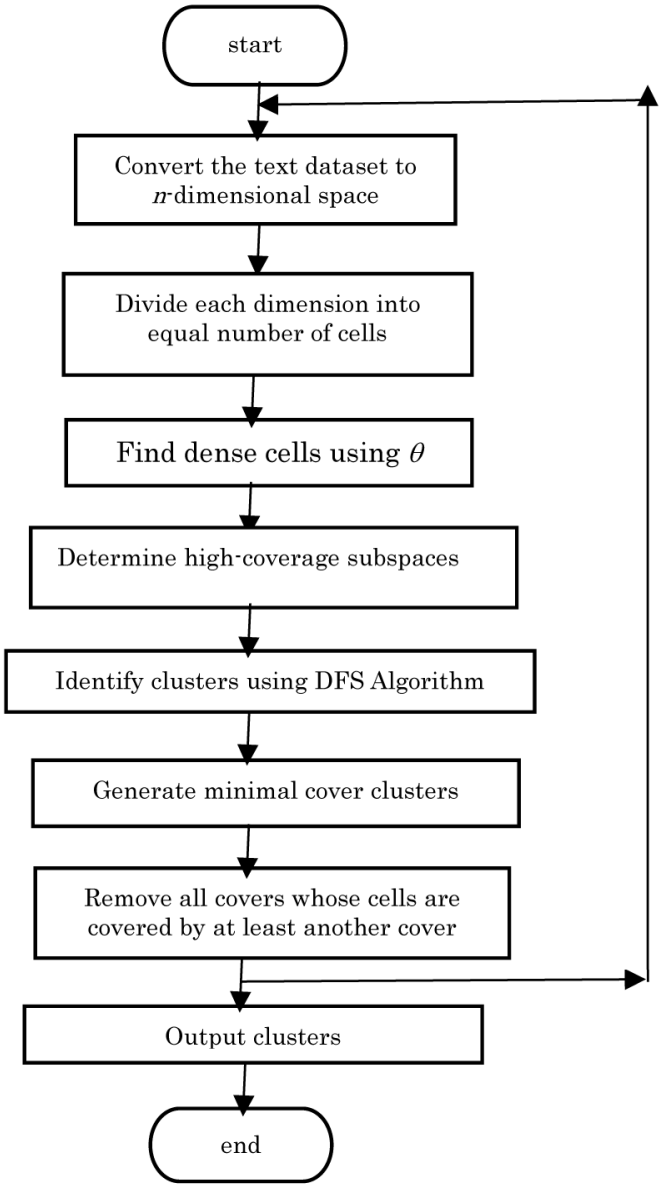

FIGURE 1. Flowchart of the algorithm

6. **Experimental Analysis and Discussions.** For conducting experiment here, we employed a dataset named bag of words [38] available on the UCI Machine Repository. The dataset contained five text collections in the form of bags of words. A detailed description of the dataset is given in Table 1.

TABLE 1. Dataset

Dataset	Dataset characteristic	Attribute characteristic	Number of instances	Number of attributes
Bag of words	Text	Integer	8000000	100000

The experiments were run on an Intel Core 7i machine with a CPU, 8 GB RAM with an MS-Windows 2010 64-bit OS. Two randomly selected samples of different sizes of instances and attributes of dataset are taken. The dimensions of the clusters and the input cluster numbers are assumed to be constant (100). Then, we run our method and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [39] with the above dataset, and the results of the execution are recorded. The obtained results are presented in Tables 2 and 3.

TABLE 2. Size of instances vs. No. of clusters

Sizes of data	Cluster's dimension	Input cluster number	Clusters by DBSCAN	Clusters by the proposed method
100000	100	100	100	100
300000	100	100	100	133
500000	100	100	110	185
800000	100	100	136	256

TABLE 3. Size of attributes vs. No. of clusters

Attribute of data	Cluster's dimension	Input cluster number	Clusters by DBSCAN	Clusters by the proposed method
100	100	100	100	100
1000	100	100	100	137
10000	100	100	123	170
100000	100	100	141	255

While running the algorithms, we first randomly selected different sizes of the dataset, i.e., 100000, 300000, 500000 and 800000, keeping the dimensions (attributes) of the dataset, the dimension of input clusters, and the numbers of input clusters are constant. We observe that our algorithm can extract more hidden clusters that cannot be extracted by DBSCAN [39]. The observations are presented in Table 2. Next, we performed a similar experiment by randomly choosing the different dimensions of the datasets while keeping other parameters constant. We make a similar observation in this case, also that our method gives more clusters than DBSCAN [39]. The observations are presented in Table 3. The results are shown in Figures 2 and 3.

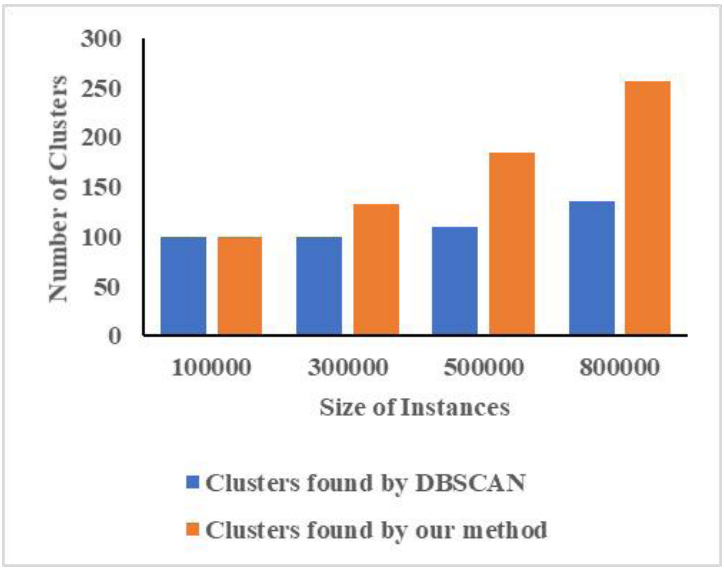

FIGURE 2. Size of instances vs No. of clusters

It is perceived from Table 2 and Figure 2, that if the size of the dataset increases, keeping the cluster's attribute or dimension and input clusters constant, the proposed algorithm gives comparatively more clusters than the DBSCAN. Similarly, if the dimension increases, keeping the dataset size and input clusters constant, the proposed algorithm gives comparatively more clusters than the DBSCAN [39]. So from the above obtained results, the following observations can be made.

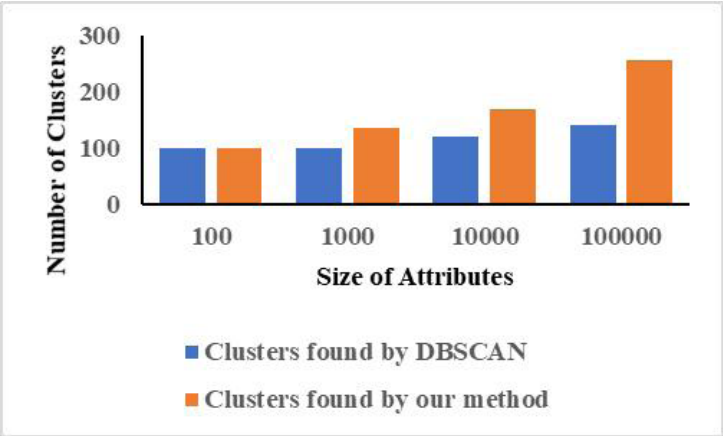

FIGURE 3. Size of attributes vs No. of clusters

- The clusters obtained by the proposed algorithm are more representative to the datasets.
- The clusters give more information about the digital or e-contents which can be easily expressed.

7. **Conclusions and Lines for Future Works.** In this article, an algorithm is discussed for clustering of documents. As the documents are semi-structured or unstructured and available in electronic form, we need to preprocess them to remove noise and undesirable contents. First, the data are converted to points in a high-dimensional space where the dimensions of the space are determined by the number of keywords. Then, an algorithm is applied to finding clusters in the subspace of the aforementioned space. The algorithm follows a density-based approach. The domain is partitioned into a given number (input parameter) of cells by dividing each dimension into equal amounts of non-overlapping intervals of the identical length. Obviously, each cell intersects one interval from one dimension. Then, each cell's density is computed by using the number of documents (or points) belonging to the corresponding cell using a user-specified input parameter. Similarly, the cell in each projection of the original n-dimensional space was specified. Using the above method, we identify a subspace that can contain clusters. Then using the DFS (Depth First Search) approach, the clusters in the subspace are found, where each cluster is a collection of connected dense cells that are maximal. These collections of connected cells are then expressed as DNF expressions. Then, a brief description of clusters is given by covering them with minimal non-redundant regions. Finally, a comparative study of the algorithm was performed using DBSCAN [39] with a real-life dataset available in the UCI machine repository, which clearly establishes the efficacy of the method in comparison to DBSCAN [39].

The following directions may be followed for future work.

- Approaches other than the aforesaid can be explored.
- Fuzzy clustering approach can be employed to deal with uncertainty or imprecision inherent in the datasets.
- Efficient algorithm can be designed to find dynamic document clusters.

Acknowledgment. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] A. K. Jain, M. N. Murty and P. J. Flynn, Data clustering: A review, *ACM Computing Surveys*, vol.31, no.3, pp.264-323, 1999.

[2] F. A. Mazarbhuiya, M. Y. AlZahrani and L. Georgieva, Anomaly detection using agglomerative hierarchical clustering algorithm, in *Lecture Notes in Electrical Engineering*, Singapore, Springer, DOI: 10.1007/978-981-13-1056-048, 2018.
[3] F. A. Mazarbhuiya, M. Y. AlZahrani and A. K. Mahanta, Detecting anomaly using partitioning clustering with merging, *ICIC Express Letters*, vol.14, no.10, pp.951-960, 2020.
[4] P. Jain, M. S. Bajpai and R. Pamula, A modified DBSCAN algorithm for anomaly detection in time-series data with seasonality, *The International Arab Journal of Information Technology*, vol.19, no.1, pp.23-28, 2022.
[5] R. A. A. Habeeb, F. Nasaruddin, A. Gani, M. A. Amanullah, I. A. T. Hashem, E. Ahmed and M. Imran, Clustering-based real-time anomaly detection – A breakthrough in big data technologies, *Transactions on Emerging Telecommunications Technologies*, vol.33, no.8, e3647, 2022.
[6] F. A. Mazarbhuiya, Detecting IoT anomaly using rough set and density based subspace clustering, *ICIC Express Letters*, vol.17, no.12, pp.1395-1403, 2023.
[7] F. A. Mazarbhuiya and M. Shenify, A mixed clustering approach for real-time anomaly detection, *Applied Sciences*, vol.13, 4151, DOI: 10.3390/app13074151, 2023.
[8] M. Steinbach, G. Karypis and V. Kumar, A comparison of document clustering techniques, *Proc. of the KDD-2000 Workshop on TextMining, Computer Science and Engineering (CS&E) Technical Reports [749]*, 2000.
[9] S. Selvaraj and E. Choi, Swarm intelligence algorithms in text document clustering with various benchmarks, *Sensors (Basel)*, vol.21, no.9, 3196, DOI: 10.3390/s21093196, 2021.
[10] R. Irfan, C. K. King, D. Grages, S. Ewen, S. U. Khan, S. A. Madani, J. Kolodziej, L. Wang, D. Chen, A. Rayes, N. Tziritas, C.-Z. Xu, A. Y. Zomaya, A. S. Alzahrani and H. Li, A survey on text mining in social networks, *The Knowledge Engineering Review*, vol.30, no.2, pp.157-170, DOI: 10.1017/ S0269888914000277, 2015.
[11] M. Shenify, Understanding user's behavior by social media data clustering, *International Journal of Advanced Trends in Computer Science and Engineering*, vol.9, no.1, pp.167-170, DOI: 10.30534/ ijatcse/2020/25912020, 2020.
[12] R. Agrawal, J. Gehrke, D. Gunopulos and P. Raghavan, Automatic subspace clustering of high dimensional data for data applications, *Data Mining and Knowledge Discovery*, vol.11, pp.5-23, 2005.
[13] International Business Machines, *IBM Intelligent Miner User's Cuide*, Version I Release 1, SH12-6213-00 Edition, 1996.
[14] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, Hoboken, 1973.
[15] M. Ester, H. P. Kriegel, J. Sandar and X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noises, *Proc. of the 2nd International Conference on Knowledge Discovery in Databases and Data Mining*, Portland, Oregon, 1996.
[16] A. Alsayat and H. El-Sayed, Social media analysis using optimized K-means clustering, *2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA)*, 2016.
[17] T. Sutanto and R. Nayak, Fine-grained document clustering via ranking and its application to social media analytics, *Social Network Analysis and Mining*, vol.8, 29, 2018.
[18] E. J. Otoo, A. Shoshani and S.-W. Hwang, Clustering high dimensional massive scientific datasets, *Journal of Intelligent Information Systems*, vol.17, no.2-3, pp.147-168, DOI: 10.1023/ A:1012853629322, 2001.
[19] J. Friedman, Optimizing a noisy function of many variables with application to mining, *UW/MSR Summer Research Institute in Data Mining*, 1997.
[20] Z. Zhou and B. Tian, Research on community detection of online social network members based on the sparse subspace clustering approach, *Future Internet*, vol.11, no.12, 254, 2019.
[21] J. Wang, A. Suzuki, L. Xu, F. Fian, L. Yang and K. Yamanishi, Orderly subspace clustering, *The 33rd AAAI Conference on Artificial Intelligence*, 2019.
[22] C. Tang, X. Zhu, X. Liu, M. Li, P. Wang, C. Zhang and L. Wang, Learning a joint affinity graph for multiview subspace clustering, *IEEE Transactions on Multimedia*, vol.21, no.7, pp.1724-1736, 2019.
[23] T. Wu, Y. Fan, Z. Hang and L. Chen, Subspace clustering on mobile data for discovering circle of friends, *Knowledge Science, Engineering and Management*, pp.703-711, 2015.
[24] A. Kaur and A. Dutta, A novel algorithm for fast and scalable subspace clustering of high-dimensional data, *Journal of Big Data*, vol.2, no.1, 17, 2015.
[25] Y. Sun and Y. Lu, A grid-based subspace clustering algorithm, *Lecture Notes in Computer Science*, vol.4256, pp.37-48, 2006.

[26] Q. Shi, B. Hu, T. Zhang and C. Zhang, Multi-view subspace clustering analysis for aggregating multiple heterogeneous omics data, *Frontiers in Genetics*, vol.10, DOI: 10.3389/fgene.2019.00744, 2019.

[27] R. V. Belfin, E. Grace Mary Kanaga and S. Kundu, Application of machine learning in the social network, in *Recent Advances in Hybrid Metaheuristics for Data Clustering*, John Wiley & Sons Ltd., 2020.

[28] A. M. Khasanova and M. O. Pasechnik, Social media analysis with machine learning, *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pp.22-27, 2021.

[29] I. A. Rytsarev, A. V. Kupriyanov, D. V. Kirsh and K. S. Liseckiy, Clustering of social media content with the use of BigData technology, *Journal of Physics: Conference Series*, vol.1096, 012085, DOI: 10.1088/1742-6596/1096/1/012085, 2018.

[30] S. Chaurasia, P. Shrivastava, M. Kamble and B. Chouksey, Study on cluster-based pattern recognition using recommendation system, *CNC 2020: Communication, Networks and Computing*, pp.121-131, 2021.

[31] M.-F. Kaya and M. Schoop, Analytical comparison of clustering techniques for the recognition of communication patterns, *Group Decision and Negotiation*, vol.31, pp.555-589, 2022.

[32] P. Sinha, L. Dey, P. Mitra and D. Thomas, A hierarchical clustering algorithm for characterizing social media users, *Companion Proceedings of the Web Conference 2020*, pp.353-362, 2020.

[33] W. Zhang and L. Pang, Multiple collaborative supervision pattern recognition method within social organizations based on data clustering algorithm, *Hindawi Journal of Mathematics*, vol.2021, pp.1-12, 2021.

[34] R. Agrawal and R. Srikant, Fast algorithms for mining association rules, *Proc. of the 20th International Conference on Very Large Data Bases*, Santiago, Chile, 1994.

[35] A. K. Mahanta, F. A. Mazarbhuiya and H. K. Baruah, Finding locally and periodically frequent sets and periodic association rules, *Proc. of the 1st International Conference on Pattern Recognition and Machine Intelligence, LNCS 3776*, pp.576-582, 2005.

[36] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, World Scientific, 1989.

[37] A. Aho, J. Hopcroft and J. Ullman, *The Design and Analysis of Computer Algorithms*, Addision Welsley, 1974.

[38] N. A. Abdulla, N. A. Mahyoub, M. Shehab and M. Al-Ayyoub, Arabic sentiment analysis: Lexicon-based and corpus-based, *IEEE Conference on Applied Electrical Engineering and Computing Technologies (AEECT 2013)*, Amman, Jordan, 2013.

[39] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, E. Simoudis, J. Han and U. M. Fayyad, A density-based algorithm for discovering clusters in large spatial databases with noise, *Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp.226-231, 1996.