axioms

MDPI

Type of the Paper (Article.)

An Improved Clustering Algorithm for Multi-Density Data

Abdulwahab Ali Almazroi [1]* and Walid Atwa [1]*

1 University of Jeddah, College of Computing and Information Technology at Khulais, Department of Information Technology, Jeddah, Saudi Arabia; *aalmazroi@uj.edu.sa*, wsatwa@uj.edu.sa

* Correspondence: *aalmazroi@uj.edu.sa*, wsatwa@uj.edu.sa;

Abstract: The clustering method divides a dataset into groups with similar data using similarity metrics. However, discovering clusters in different densities, shapes and distinct sizes is still a challenging task. In this regard experts and researchers opt to use the DBSCAN algorithm as it uses density-based clustering techniques that define clusters of different sizes and shapes. However, it is misapplied to clusters of different densities due to its global attributes that generate a single density. Furthermore, most existing algorithms are unsupervised methods where available prior knowledge is useless. To address these problems, this research suggests the use of a clustering algorithm that is semi-supervised. This allows the algorithm to use existing knowledge to generate pairwise constraints for clustering multi-density data. The proposed algorithm consists of two stages, first, it divides the dataset into different sets based on their density level and then applies the semi-supervised DBSCAN algorithm on each partition. Evaluation results show the algorithm performing effectively and efficiently in comparison to unsupervised clustering algorithms.

Keywords: Clustering; DBSCAN; Pairwise Constraints; Multi-Density Data.

1. Introduction

Clustering is utilized to arrange the dataset to a limited set of clusters based on selected similarity metrics [1]. There are different clustering categories like density-based algorithms that can identify clusters of distinct sizes and shapes. Hence, this algorithm does not require the specification of clusters number. Rather, clusters are identified in a densely connected region which grows based on the direction of density. Meaning, that density-based algorithms identify clusters based on regions that have a high density and are separated from the regions with low densities [2-5].

The DBSCAN algorithm forward all advantages of the density-based clustering family [6]. It computes the density by counting the number of points within a circle with a given radius (referred to as *Eps*) surrounding the point. Core points are characterized as having a density over a predetermined threshold (called *MinPts*) [6]. The two provided parameters (*Eps* and *MinPts*) define a single density. Thus, the clustering methods based on the DBSCAN can't perform well with multi-density data [7, 8]. Moreover, most of these methods are unsupervised, and can't utilize the prior knowledge that can be available either in the form of labeled data or pairwise constraints.

Semi-supervised clustering improves the clustering performance with the pairwise constraints or labeled data. The pairwise constraints are of two types: *must-link* and *cannot-link* constraints. A *must-link* constraint (i.e. $ML(x, y)$) means that the two objects x and y must be in the same cluster. While, the *cannot-link* constraint (i.e. $CL(x, y)$) means that these two objects x and y must be in different clusters.

In this study, we present a semi-supervised clustering algorithm for clustering multi-density data, referred to as the SSMD. Our algorithm is separated into two core parts. Firstly, we divide the data into various density levels and calculate the density parameters for each density level set. Secondly, we apply the pairwise constraints to get the set of

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Axioms* **2022**, *11*, x. https://doi.org/10.3390/xxxxx

Academic Editor: Firstname Lastname

Received: date
Accepted: date
Published: date

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

clusters based on the computed parameters. We conduct experiments over real datasets. Comparison results reveal the effectiveness and efficiency of the proposed algorithm.

The main contributions are twofold. Firstly, our proposed algorithm can discover clusters of changing densities using the available pairwise constraints. Secondly, we evaluate the performance proposed algorithm on different experiments with varied datasets in comparison to other algorithms.

The remaining part of the paper is organized as follows. The related work is explained in Section 2. In Section 3, SSMD algorithm is discussed in details. Section 4 shows the performance analysis of the SSMD algorithm compared with other algorithms. Lastly, the paper is concluded in Section 5.

2. Related Work

In this section, key literature related to clustering methods are highlighted and discussed. Hence, literature on density-based clustering and semi-supervised clustering algorithms are critically reviewed.

For the purpose of finding clusters in huge spatial data sets, Ester et al. proposed a density-based clustering algorithm named DBSCAN [6]. The DBSCAN algorithm depends on the two specified parameters (*Eps* and *Minpts*) that define a single density. Thus, the DBSCAN can't cluster datasets well with large differences in densities.

Ankerst et al. proposed an algorithm named OPTICS that executes and stores dual parameters which are core distance and reachability distance for cluster identification with different densities [9]. If the *Eps*-neighborhood of a point p contains at least *MinPts* points, then that point is a core point. As a result, a core distance is assigned to each point, describing how far it is from the *MinPts-th* closest point. The greater of the distances between two points, o and p, or p's core distance, defines the reachability distance between them.The algorithm further creates an ordering process for the dataset which will represent its density-based clustering structure. The study reveals the process of extracting not both traditional clustering information and intrinsic clustering structure automatically with good efficiency [9].

Based on the issue of identifying clusters in high-dimensional data, Ertoz et al. proposed an algorithm that identify clusters in distinct sizes, shapes, and densities [10]. Firstly, the algorithm check for the nearest neighbors (*NN*) of individual data points and further describes the similarity that resides between the points with respect to the number of *NN* shared by the points. Hence, the algorithm defines and builds clusters around these defined points. An experiment on various datasets shows that the algorithm achieves good performance compared to traditional methods. To detect clusters of distinct shapes and sizes, Liu et al. proposed an altered version of the DBSCAN algorithm coined the term Entropy and Probability Distribution (EPDCA) [11] for it. Testing on benchmark datasets indicates that clustering results based on EPDCA achieve good performance.

A study by Kim et al. proposed a density-based clustering algorithm coined approximate adaptive AA-DBSCAN [12]. The algorithm focused on minimizing extra computation needed to determine parameters by utilizing e-distance for each density when identifying clusters. Based on an experiment conducted, the result shows a significant improvement with regards to clustering performance with a decrease in running time. Another study by Zhang et al., proposed GCMDDBSCAN because DBSCAN can't handle databases that are large [13]. The authors highlighted that the clustering capabilities on datasets that are large have improved together with clustering accuracy.

FlockStream was proposed by Forestiero et al. The algorithm is based on a multi-agent system and the results demonstrate that FlockStream has good performance on both synthetic and real datasets [14]. Chen and Tu proposed a framework named D-Stream with the aim of clustering stream data by utilizing a density-based approach [15].

Recently, semi-supervised clustering algorithms [16-22] are an extension of known unsupervised clustering algorithms that utilize the background knowledge in the form of labeled data or pairwise constraints to improve the clustering performance. Huang et al.

proposed an algorithm named MDBSCAN that handles multi-density datasets by automatically calculating the parameter *Eps* for each density distribution using the pairwise constraints [16].

A study by Ruiz et al. proposed a pairwise constrained clustering algorithm (C-DBSCAN) that utilizes the information pairwise constraints to enhance clustering performance [17]. C-DBSCAN creates a set of neighborhoods based on cannot-link constraints and uses must-link constraints to merge the local clusters in each neighborhood. The result reveals that C-DBSCAN can detect arbitrary shapes and evolving clusters with current pairwise constraints. In another study by Lelis et al., a new density-based semi-supervised clustering algorithm was proposed (SSDBSCAN) that utilizes the labeled data for evaluating the density parameters [18]. The result shows an improvement in clustering accuracy with little supervision required. However, C-DBSCAN and SSDBSCAN can't apply well with the multi-density data, especially with increasing the number of pairwise constraints.

Wagstaff et al. proposed a well-known semi-supervised clustering algorithm named MPCKmean. The MPCKmean is a variant of *K*-means algorithm that use the pairwise constrints for clustering [19]. It is very effective at processing huge datasets, but it can't handle clusters of different sizes and densities.

3. SSMD Algorithm

Semi-supervised clustering algorithms utilize a set of class label constraints on some examples to help unsupervised clustering. We propose an active semi-supervised clustering for a multi-density dataset (called the ***SSMD***) that attempts to find out clusters with distinct densities and arbitrary shapes. The *SSMD* algorithm has two main components: (1) Partition the dataset *D* into distinct density level sets. (2) Apply the *Semi-DBSCAN* algorithm on each density level set where *Semi-DBSCAN* is an extension of *DBSCAN* that makes full use of existing pairwise constraints. The pseudocode of our *SSMD* is revealed in Algorithm 1.

First, we compute the density function for each data point (p) as shown in Eq (1). Then sorting the density for all the data points in descending order to evaluate the variation between each sequence point. Density variation explains how much denser or sparser the points are.

$$Den(p, MinPts) = \frac{kdist(p,k)}{k}, \quad (1)$$

We define $kdist(p, k)$ as the number of p's neighbors and k is an arbitrary positive integer that explains k-nearest neighbor distance. The density variation of point p_i with respect to p_j is computed in Eq (2).

$$DenVar(p_i, p_j) = \frac{\left|Den(p_i,k) - Den(p_j,k)\right|}{Den(p_i,k)}. \quad (2)$$

We partition the dataset into a list of density level sets (*DLS*). The *DLS* contains a set of data points whose densities are approximately the same. We compute the density variation threshold (τ) according to the statistical characteristics of the sorted density variation list.

$$\tau = \frac{\sum_{i=0}^{n} DenVar_{(i)}}{n} \; where \;\; DenVar_{(i)} > 0. \quad (3)$$

Finally, we separate the data points into different densities based on the computed density variation threshold (τ) as follows:

$$p_i, p_j \in DLS_k \;\; if \; DenVar(p_i, p_j) \leq \tau. \quad (4)$$

After density levels partitioning, we acquire a list of density level sets (for short *DLSList*). Then we need to find representative *Eps* for each density level set. We compute *Epsi* for DLS_i as follows:

$$Eps_i = \max(DLS_i) \cdot \sqrt{\frac{median(DLS_i)}{mean(DLS_i)}}. \quad (5)$$

We define *max*, *median*, and *mean* as the maximum, median, and mean values of each DLS_i respectively.

Algorithm 1 SSMD

Input:

Dataset (*D*), must-link and cannot-link constraints (*ML*, *CL*), number of objects in a neighborhood (*MinPts*)

Output:

Set of clusters

Begin

Compute density value for all data points according to *eq(1)*

Sort density list in descending order

Compute Density variation values using *eq(2)*

Generate Density Level Set (*DLS*) using a threshold in *eq(3)*

EpsList = EstimateEps(DLSList);

For each DLS_i in *DLSList*

Semi-DBSCAN(DLS_i, *ML*, *CL*, *MinPts*, Eps_i)

End For

Return all clusters

End

With density levels partitioning and parameters estimation done, we carry on the clustering process: Initialize global parameter *MinPts* = *k*, adopt *Semi-DBSCAN* algorithm for each Eps_i in *EpsList* on corresponding *DLS*. The algorithm s*emi-DBSCAN* can find a set of clusters and a (probably empty) set of noises. Initially, the constraints are preprocessed. Constraints given by human experts may be incomplete. Some constraints are not explicitly given, for example, if we have a $ML(x_i, x_j)$ and $ML(x_i, x_k)$ then we have $ML(x_j, x_k)$. Then, we use constraints to monitor the process of growing clusters in Semi-DBSCAN as presented in Algorithm 2.

First, we compute the neighborhood for each unclassified data point and compare it with *MinPts* to determine whether it can be added to the current cluster or noise set. Then, expanding a cluster using the pairwise constraints as follows:

1. Add all data points that have a must-link constraint with *d* to the current cluster.
2. Add all data points in *d's* neighborhood that do not violate the cannot-link constraint with *d* to the current cluster.

Algorithm 2 Semi-DBSCAN (*D, ML, CL, MinPts, Eps*)

Begin

Cluster.Id: = 0;

For each *d* in *D*

If *d* is *UNCLASSIFIED* then

Compute *d's Eps*-neighborhood *neighborhood*;

a. **If** *neighborhood* < *MinPts*

Add d to NOISE set;

Else

Add d to current cluster (*ClusterId*);

b. **For** each data point *o* has a must-link constraints *ML*(*d*, *o*)

Add o to current cluster (*ClusterId*);

c. **For** each data point *p* in the neighborhood set that does not violate cannot-link constraints

Add p to current cluster (*ClusterId*);

ClusterId = *ClusterId* +1;

End

4. Experimental Results

In this section, the datasets used are presented, followed by the evaluation metrics utilized, and the results of the study.

4.1. Datasets and Evaluation Metrics

In this section, the datasets utilized are presented. Table 1 outlines the used datasets. For our experiments, real datasets like Magic and Glass were selected from the UCI repository. Hence, these datasets are labeled with instances, attributes, and cluster labels as described in Table 1.

Table 1. The Data Sets Used in the Experiments.

Dataset	# Instances	#Attributes	#Clusters
Glass	214	10	6
Ecoli	336	8	8
Segment	2310	19	7
Magic	19020	10	2

We utilized four clustering algorithms, i.e., C-DBSCAN [17], SSDBSCAN [18], AA-DBSCAN [12], and MPCKmean [19], along with SSMD to compare their clustering performances. Clustering performance was measured in terms of normalized mutual information (*NMI*) and Clustering Accuracy (*ACC*). *NMI* and Clustering accuracy is measured using Equations 6 and 7.

$$NMI = \frac{I(X;Y)}{\big(H(X) + H(Y)\big)/2}. \quad (6)$$

$$ACC = \frac{\sum_{i=1}^{n} I(\hat{t}_i = t_i)}{n}. \quad (7)$$

4.2. Performance Analysis

In this sub-section, the performance evaluation based on NMI is presented. This evaluation result is for the proposed algorithm in comparison to the four algorithms (SSDBSCAN, AA-DBSCAN, MPCKmeans, and C-DBSCAN). The goal is to find out if or not the proposed algorithm performs better with varied constraints on distinct datasets. Hence, this experiment will present evidence that the proposed algorithm is more effective in comparison to the compared algorithms.

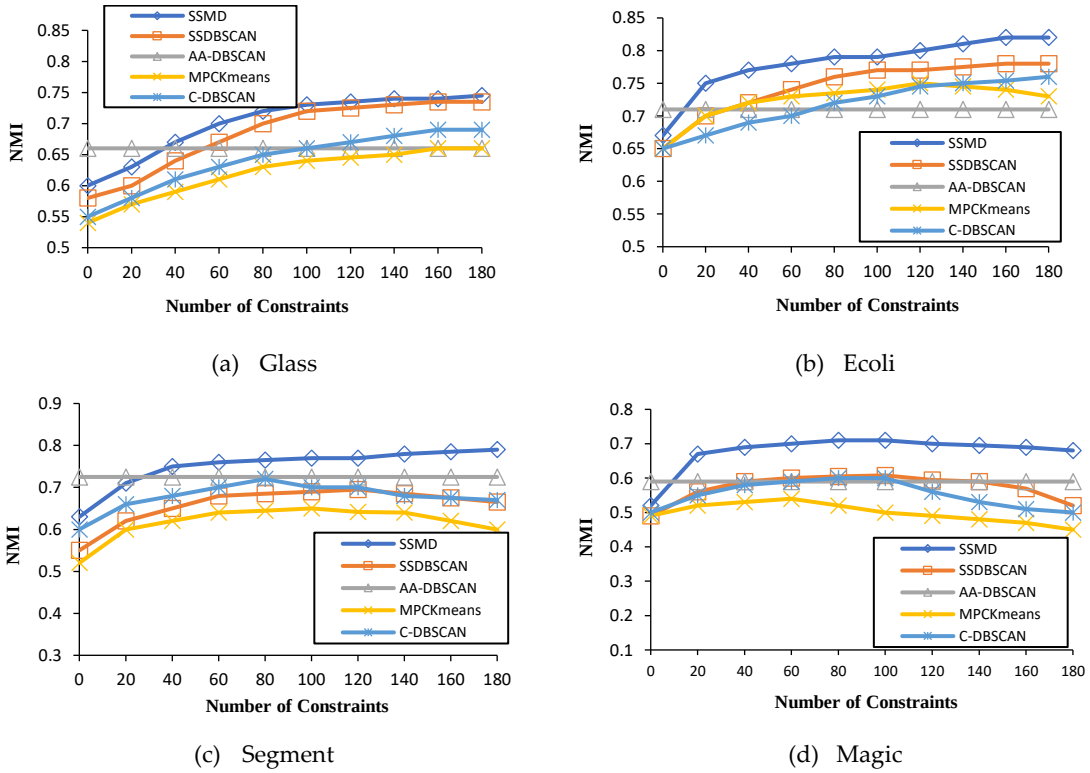


Figure 1. Comparison of NMI.

Figure 1 presents the performance results of the four different datasets (Glass, Ecoli, Segment, and Magic) as presented in Table 1. For each dataset, several constraints were utilized for the NMI.

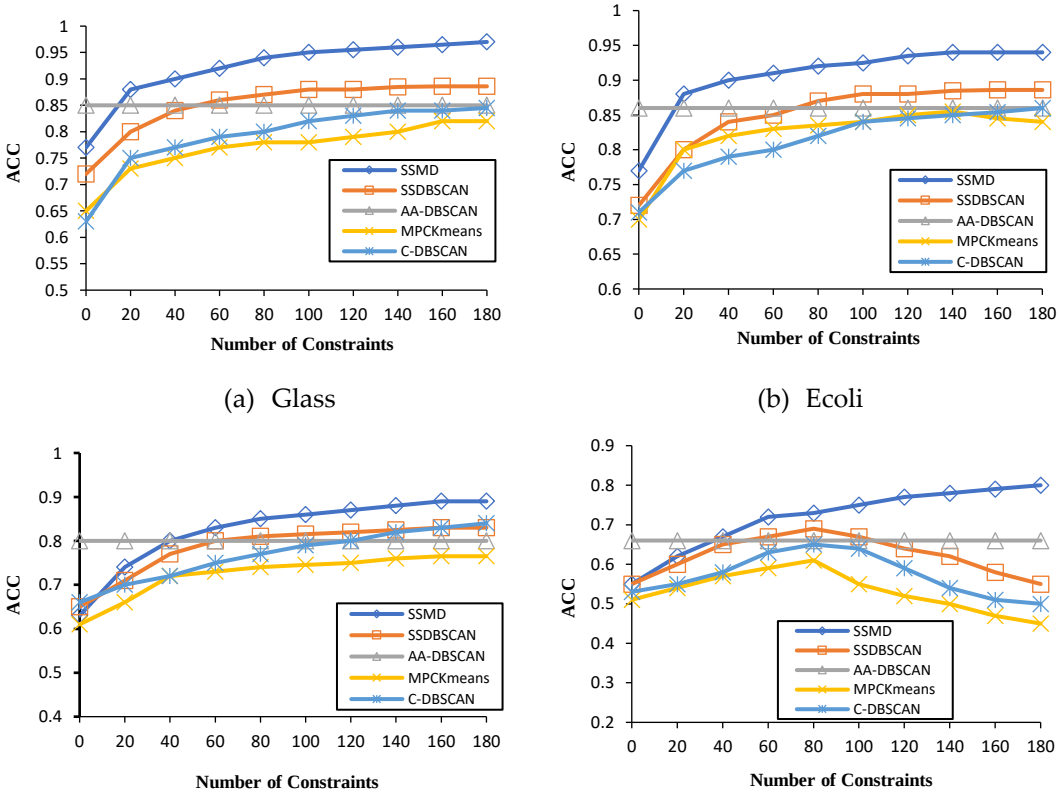
It can be observed from Figure 1, that SSMD algorithm generally performs better than the four other methods when the number of constraints increased (e.g. ecoli, segment, and magic). For instance, in Figure 1(a), with 20 constraints the proposed algorithm performance is superior with greater than 0.6 NMI, and with 80 constraints, the proposed algorithm achieves NMI of more than 0.7, respectively. Looking very carefully, one can see that SSDBSCAN is the second most effective, followed by C-DBSCAN. It can be seen from Figure 1, that the performance of AA-DBSCAN in all datasets is constant value as it is unsupervised clustering algorithms. Thus, this prove the utility of semi-supervised clustering algorithms over unsupervised approaches when knowledge are available.

For the Ecoli dataset, the SSMD algorithm has also shown to be more effective, with 0.8 NMI with 80 constraints. This is the same with the other two datasets, which are Segment and Magic. Considering the overall result in Figure 1, we can confidently conclude that for NMI performance evaluation, the proposed algorithm outperformance all the four compared algorithms.

Also, we present the results based on the clustering accuracy of the SSMD algorithm in comparison to compared algorithms on the datasets outlined in Table 1. Hence, the results of each algorithm are displayed in varied colors as shown in Figure 2. From Figure 2, the proposed algorithm SSMD has a much higher accuracy and more stable state than SSDBSCAN and C-DBSCAN. For instance, looking at Figure 2(a), the accuracy of the SSMD algorithm reaches 88% when selecting 20 constraints from all constraints, and reach the highest accuracy when selecting more than 100 constraints.

Considering the general clustering performance on all datasets in Figure 2, SSDBSCAN is the second most effective algorithm for clustering accuracy on Glass, Ecoli, and Segment datasets, followed by AA-DBSCAN and C-DBSCAN. The main advantage of the proposed algorithm is that it can attain good performance with fewer constraints and

progressively maintain the performance. However, we observed that other algorithms compared required more constraints to get a good performance. Some algorithms' performance also subsides considerably.



(a) Glass (b) Ecoli

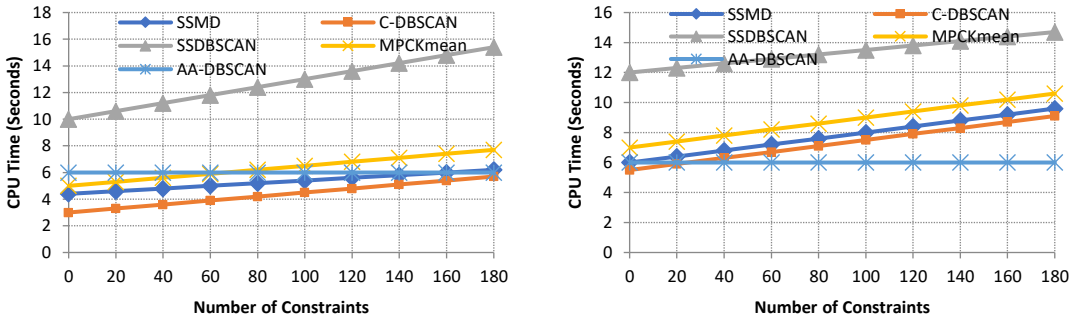
(c) Segment (d) Magic

Figure 2. Comparison of the clustering accuracy.

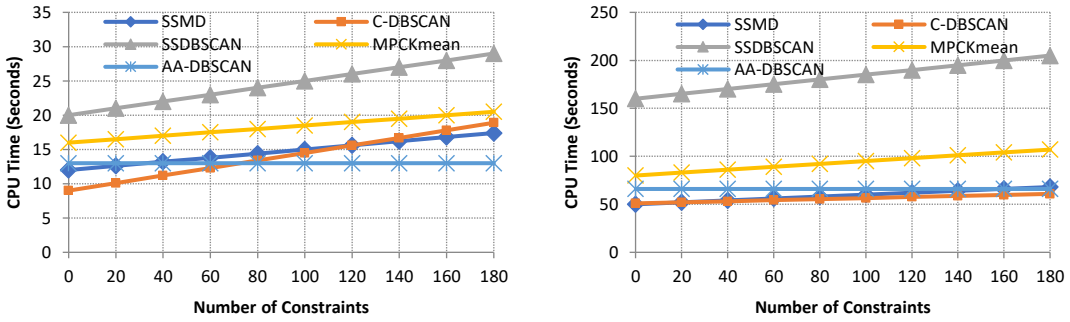
4.3. Efficiency Analysis Based on Times

In this section, the clustering times for the proposed algorithm and the four compared algorithms are presented. Figure 3 gives the results for the execution times using a distinct number of pairwise constraints on the four datasets. Achieving a low executing time indicates that an algorithm has better performance, and vice versa with a high execution time. From Figure 3(a) SSDBSCAN has the highest execution time with all pairwise constraints taken into consideration.

It can be seen that the SSMD method is generally time efficient (about 5 seconds on the glass dataset with 214 samples, and about 50 seconds on the magic dataset with 19020 samples possessing). Even though the SSMD isn't always the fastest method for all of the data sets, it performs substantially better than its competitors.



(a) Glass (b) Ecoli



(c) Segment (d) Magic

Figure 3. Comparison of the execution time.

5. Conclusions

In this paper, the SSMD algorithm is a semi-supervised clustering algorithm for clustering multi-density data and arbitrary shapes. The proposed algorithm partitions the dataset into distinct density level sets by examining the statistical characteristics of its variation with respect to density, and then expands the clusters using selected active pairwise constraints. We conducted experiments to assess the clustering performance and execution time on both artificial and real datasets of distinct dimensions and sizes. The experimental result revealed that the SSMD attain better clustering performance in comparison to the existing state-of-the-art. Furthermore, the SSMD has significantly reduced the execution time in comparison to the compared algorithms in most scenarios.

Acknowledgments: This work was funded by the University of Jeddah, Saudi Arabia, under grant No. (UJ-02-069-DR). The authors, therefore, acknowledge with thanks the university's technical and financial support.

References

1. T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm: Analysis and implementation,”. IEEE transactions on pattern analysis and machine intelligence, vol 24, no. 7, pp. 881-892, 2002.
2. Lulli, M. Dell’Amico, P. Michiardi, and L. Ricci, “NG-DBSCAN: scalable density-based clustering for arbitrary data,” Proc. VLDB Endow., vol. 10, no. 3, pp. 157–168, 2016.
3. A. Latifi-Pakdehi and N. Daneshpour, DBHC: A DBSCAN-based hierarchical clustering algorithm. Data & Knowledge Engineering, 135, p.101922, 2021.
4. Y. Yang, C. Qian, H. Li, Y. Gao, J. Wu, C. J. Liu, and S. Zhao. "An efficient DBSCAN optimized by arithmetic optimization algorithm with opposition-based learning." The Journal of Supercomputing 1-39, 2022.
5. M. Li, X. Bi, L. Wang, and X. Han. "A method of two-stage clustering learning based on improved DBSCAN and density peak algorithm." Computer Communications 167 pp 75-84, 2021.
6. M. Ester, H.P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise”. In Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231, 1996.

7. W. Atwa, and A. A. Almazroi, "Active selection constraints for semi-supervised clustering algorithms". Int. J. Inf. Technol. Comput. Sci.(IJITCS), 12(6), 23-30, 2020.
8. R. Qasim, W. H. Bangyal, M. A. Alqarni, and A. A. Almazroi, "A fine-tuned BERT-based transfer learning approach for text classification". Journal of healthcare engineering, 2022.
9. M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," ACM Sigmod Rec., vol. 28, no. 2, pp. 49–60, 1999.
10. L. Ertöz, M. Steinbach, and V. Kumar, "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data," in Proceedings of the 2003 SIAM international conference on data mining, 2003, pp. 47–58.
11. X. Liu, Q. Yang, and L. He, "A novel DBSCAN with entropy and probability for mixed data," Cluster Comput., vol. 20, no. 2, pp. 1313–1323, 2017.
12. J.-H. Kim, J.-H. Choi, K.-H. Yoo, and A. Nasridinov, "AA-DBSCAN: an approximate adaptive DBSCAN for finding clusters with varying densities," J. Supercomput., vol. 75, no. 1, pp. 142–169, 2019.
13. L. Zhang, Z. Xu, and F. Si, "GCMDDBSCAN: multi-density DBSCAN based on grid and contribution," in 2013 IEEE 11th International Conference on Dependable, Autonomic and Secure Computing, pp. 502–507, 2013.
14. A. Forestiero, C. Pizzuti, and G. Spezzano, "A single pass algorithm for clustering evolving data streams based on swarm intelligence," Data Min. Knowl. Discov., vol. 26, no. 1, pp. 1–26, 2013.
15. Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007, pp. 133–142.
16. T. Huang, Y. Yu, K. Li, and W. Zeng, "Reckon the parameter of DBSCAN for multi-density data sets with constraints," in 2009 International Conference on Artificial Intelligence and Computational Intelligence, 2009, vol. 4, pp. 375–379.
17. C. Ruiz, M. Spiliopoulou and E. Menasalvas. C-DBSCAN: Density-Based Clustering with Constraints. In Proceedings of the International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, pages 216- 223, 2007.
18. L. Lelis and J. Sander. Semi-Supervised Density-Based Clustering. In Proceeding of the Ninth IEEE International Conference on Data Mining, 842-847, 2009.
19. K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in ICML, vol. 1, pp. 577–584, 2001.
20. D. M. Ablel-Rheem, A. O. Ibrahim, S. Kasim, A. A. Almazroi, and M. A. Ismail, "Hybrid feature selection and ensemble learning method for spam email classification". International Journal of Advanced Trends in Computer Science and Engineering, 9(1.4), pp 217-223, 2020.
21. M. A. Masud, J. Z. Huang, M. Zhong, and X. Fu, "Generate pairwise constraints from unlabeled data for semi-supervised clustering," Data Knowl. Eng., vol. 123, p. 101715, 2019.
22. W. Atwa, "A Supervised Feature Selection Method with Active Pairwise Constraints." In Proceedings of the 11th International Conference on Informatics & Systems, 2018.