MAFIA: A Study of Subspace Clustering Algorithms

Henrik Daniel Christensen[*hench13@student.sdu.dk*]

University of Southern Denmark, SDU
Department of Mathematics and Computer Science

Abstract. This paper presents a comparative analysis of three subspace clustering algorithms: CLIQUE, MAFIA, and SUBCLU with a main focus on MAFIA. Their scalability, clustering quality, and sensitivity to parameters is investigated. MAFIA, with its adaptive grid approach, demonstrates the highest scalability for large datasets. CLIQUE, while also scalable, struggles with clustering quality, especially in complex data where grid size is not well-tuned. SUBCLU, utilizing a density-connected approach, excels at identifying arbitrarily shaped clusters but incurs a higher computational cost. The study underscores that all three need proper parameter tuning and the choice of algorithm depends on the specific requirements and use case.

Keywords: High-Dimensional Subspace Clustering · MAFIA · Grid-Based and Density-Connected approach · Comparative Study.

1 Introduction

Clustering is one of the main techniques within data mining. The main goal is to discover unknown patterns within a data set, by partitioning the data objects into *clusters* in reasonable amount of time. Here, each object in a cluster is similar to one another, but different from objects in other clusters. [6, p. 444].

As the data sets, today, often are large in size, the clustering algorithm must be *scalable*. Additionally, data sets often contains numerous features/dimensions (*attributes*), which introduces the problem of *curse of dimensionality*, which refers to mainly two challenges: (1) *concentration of distances*, where distances between objects in high-dimensional spaces become increasingly similar as dimensionality increases. This means that data points tend to become nearly equidistant from one another, making it difficult for traditional distance-based clustering algorithms to discover clusters. (2) *feature relevance* and *local feature correlation*, where only a subset of features or different combinations of feature correlations may be relevant for clustering. Consequently, feature reduction techniques like *Principal Component Analysis* (PCA), which project the original space onto a lower-dimensional subspace, are inadequate because they typically identify only one global subspace that fits best. Also, algorithms that evaluate the entire feature space does not address this issue effectively. [9, p. 43–46]

Therefore, a local approach essential to address the second problem. However, this introduces two separate problems that the clustering algorithm must solve

simultaneously: (1) identifying the relevant subspaces for each cluster, and (2) discovering the clusters within those relevant subspaces. To solve them efficiently, heuristics needs to be employed into the clustering algorithm. [9, p. 6–7]

A common heuristic used is to focus on clusters within axis-parallel subspaces, which limits the search space to $O(2^d)$ attributes. Algorithms that follow this strategy are known as *subspace clustering* (or *projected clustering*) algorithms. These algorithms follows one of two approaches: the *top-down* or the *bottom-up* approach. In the top-down approach, the relevant subspaces for the clusters are determined by gradually reducing the subspaces, starting from the entire space. In contrast, the bottom-up approach, finds the relevant subspaces for the clusters from the original space starting from 1-dimensional using the *monotonicity property* (or *downward closure property*), see Lemma 1 [2]. [9, p. 8, 11]

For the remainder of this paper, the following notation will be used: Let $\mathcal{A} = \{A_1, \ldots, A_d\}$ represent the set of *attributes* in a d-dimensional numerical space, and let $\mathcal{S} = A_1 \times A_2 \times \cdots \times A_d$ be the full attribute space. A *subspace* is then defined as any subset of $\mathcal{A}$. Let $\mathcal{D}$ denote the *dataset*, and a *cluster* $\mathcal{C}$ is a subset of $\mathcal{D}$.

Lemma 1. *If $\mathcal{C}$ is a cluster in a k-dimensional subspace, then $\mathcal{C}$ must also form a cluster in any $(k-1)$-dimensional projection of the same space.*

A proof is provided in [2].

1.1 Contributions

This paper focuses on analyzing the grid-based bottom-up subspace clustering algorithm MAFIA [12], which extends the earlier grid-based approach CLIQUE [2]. The relationship between these two algorithms is closely examined. Additionally, the density-connected algorithm SUBCLU [10] is included as a contrasting method to the grid-based approaches. All three algorithms are evaluated experimentally, with a focus on scalability and clustering quality.

The remainder of the paper is organized as follows: Section 2 provides a detailed description and analysis of the three algorithms. Section 3 presents an evaluation of their performance, focusing on scalability with respect to dataset size, data- and cluster-dimensionality, and clustering quality. Section 4 discusses the results and the methodologies used. Finally, Section 5 concludes with the main findings and proposes future work.

2 Subspace Clustering

The overall idea of subspace clustering is to identify the subspaces that allows better clustering than the full space $\mathcal{S}$. This section discusses two different approaches to subspace clustering, including an analysis of three specific algorithms. All three algorithms follow a bottom-up approach, which requires them to implement the monotonicity property [9, p. 1:11], as outlined in Lemma 1.

2.1 Grid-Based approach

In a grid-based approach, $\mathcal{S}$ is partitioned into an axis-parallel grid structure, where each grid cell forms a axis-aligned, hyper-rectangular *unit*. The number of points within each unit is then counted. This process begins in the 1-dimensional space for each attribute. Units containing a number of points exceeding a predefined *density threshold* are retained as *dense units* (DUs). These DUs are then combined with adjacent ones in the next level (2-dimensional space) to form *candidate dense units* (CDUs). CDUs that meet the density treshold are kept, and this proces continues iteratively. The final set of DUs represent the clusters discovered by the algorithm. These clusters are typically described using minimal representations in the form of *Disjunctive Normal Form* (DNF) expressions, defined by intervals of the attributes. Two algorithms that follows this grid-based approach are CLIQUE and MAFIA.

CLIQUE is the first subspace clustering method to use a grid-based approach. It begins by partitioning $\mathcal{S}$ into equal-sized *windows* (or *units*) with a width of ε (input parameter). In the 1-dimensional case, DUs are identified by counting the number of points in each window, often using a histogram, as shown in Figure 1, where each window correspond to a specific *bin* count. For example, if $\varepsilon = 0.2$, CLIQUE identifies 3 DUs in A_1 and 4 DUs in A_2. The total of 7 DUs in the 1-dimensional space is then used to form CDUs in the 2-dimensional space. Specially, CLIQUE uses the following definition to form CDUs in the k-dimensional space:

Definition 1. *CDUs in k-dimensional space are formed by merging DUs from the $(k-1)$-dimensional space that share the first $(k-2)$ attributes.*

måske gøre det klart at det ikke har noget med Apriori at gøre.

We refer to Definition 1 as the *first-approach*. Once the CDUs are formed, only those that are actually dense are retained for further processing.

To optimize performance, CLIQUE applies a pruning technique by calculating the *coverage* of subspaces. Subspaces with low coverage–those containing only a small number of points–are pruned to reduce computation time. This method helps manage the potentially large number of CDUs generated by CLIQUE. However, this pruning strategy carries the risk of discarding subspaces that may contain important cluster information.

Once the final DUs are identified, CLIQUE groups them into clusters. A *region* is formed by combining adjacent DUs within a subspace. A *maximal region* is the largest set of connected DUs that cannot be expanded further by adding more adjacent DUs. These maximal regions define the extent of a cluster within the given subspace. For instance, in Figure 1, two distinct maximal regions, labeled A and B, are shown. The minimal representation of the cluster in DNF is: $((0.2 \leq A_1 < 0.6) \wedge (0.4 \leq A_2 < 0.8)) \vee ((0.4 \leq A_1 < 0.8) \wedge (0.2 \leq A_2 < 0.6))$.

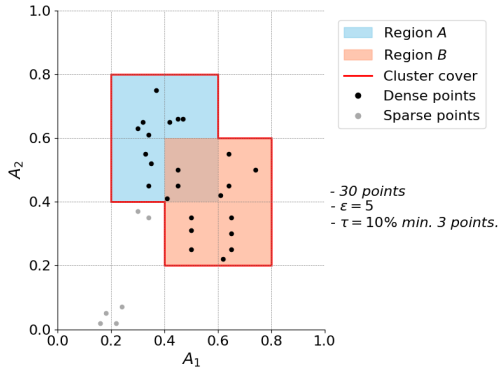


Fig. 1. 2-dimensional space containing 2 overlapping dense regions with 8 DUs in total.

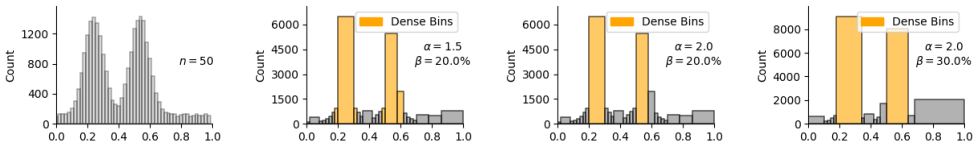
MAFIA extends CLIQUE by introducing the following modifications:

1. Automatically determines variable-sized grids based on the data distribution.
2. Forms CDUs using all combinations of the dimensions present in the DUs.
3. Eliminates the pruning technique to avoid potential loss of important cluster information, as previously noted.
4. Attempts to parallelize the clustering process to enhance performance.

Point 1 and 2 will in the following be explained in more detail.

Adaptive Grid Algorithm. MAFIA begins by partitioning each attribute into many (n-number) of equal-sized windows (default $n = 1000$). Then, from left to right, two adjacent windows are merged if their bin count differ by less than a predefined percentage, β (input parameter). A higher β value results in more windows being merged, and vice-versa. When two windows are merged, the highest bin count of the two is carried formward for comparison with the next window, rather than summing the counts of the merged pair. Figure 2 illustrates examples before and after applying the adaptive algorithm with different α and β values.

After merging, MAFIA determines a *dense-level* for each window, which establishes how many points the window must contain to be considered dense and thus candidate for forming CDUs in higher-dimensional spaces. The dense-level is calculated as: dense-level $= \frac{\alpha \cdot a \cdot |\mathcal{D}|}{D_i}$, where a is the *size* of the window, α is the *cluster dominance factor* (input parameter), and D_i is the total range of A_i. A higher α value leads to a higher dense-level, requiring more points for a window (i.e. unit) to qualify as dense, as shown in Figure 2.



(a) Before Adaptive. **(b)** After Adaptive, $\alpha = 1.5$, $\beta = 20\%$. **(c)** After Adaptive, $\alpha = 2$, $\beta = 20\%$. **(d)** After Adaptive, $\alpha = 2$, $\beta = 30\%$.

Fig. 2. Illustration of the effect of adaptive grids.

CDU Generation in MAFIA. MAFIA's approach to generating CDUs differs slightly from that of CLIQUE. MAFIA uses the following definition for forming CDUs in k-dimensional space:

Definition 2. *CDUs in k-dimensional space are formed by merging DUs from the $(k-1)$-dimensional space that share any $(k-2)$ attributes.*

We refer to Definition 2 as the *any-approach*. The key distinction between CLIQUE's *first-approach* and MAFIA's any-approach is that MAFIA considers any combination of dimensions shared by the DUs, rather than focusing only on a fixed set of dimensions. Table 1 illustrates the advantage of the any-approach. Given 3 DUs in a 4-dimensional space, both CLIQUE and MAFIA would form a 4-dimensional CDU from DU1 and DU2, as they share the first two dimensions. However, only MAFIA would also form a CDU from DU1 and DU3.

	DU1	DU2	DU3
Dim. used by DU	$\{1,2,3\}$	$\{1,2,4\}$	$\{2,3,4\}$

Table 1. Illustration showing the difference between the *first-approach* and *any-approach* with three 3-dimensional DUs.

While the any-approach can lead to a large number of combinations, potentially resulting in a worst-case time complexity of $O(Ndu^2)$, where Ndu is the number of DUs, MAFIA mitigates this through its adaptive grids, which reduce the number of DUs generated. Although CLIQUE could theoretically implement the any-approach, larger grid sizes would definitely be required to limit the number of DUs, making this less efficient. Note that, the any-approach can lead to duplicate CDUs, which must be eliminated before progressing to higher dimensions.

2.2 Density-Connected approach

A drawback of grid-based methods is that clustering quality depends on grid positioning, as illustrated in Figure 1. Additionally, clusters that deviate from hyper-rectangular shapes are challenging to detect. A potential solution is to use a density-connected algorithm like SUBCLU.

SUBCLU extends the principles of the well-known *DBSCAN* algorithm [3] to higher dimensions. It begins by applying DBSCAN in each 1-dimensional subspace separately, identifying dense regions using two input parameters: ε, which defines the neighborhood radius, and $minPts$, the minimum number of points required for a point to qualify as a *core point*. Points within ε of a core point, but not dense enough to be core points themselves, are classified as *border points*, while all remaining points are treated as *noise*.

A *density-connected set* consists of core points that are connected through other core points, along with border points that extend the cluster but do not meet the density criteria. These border points are connected to the cluster via

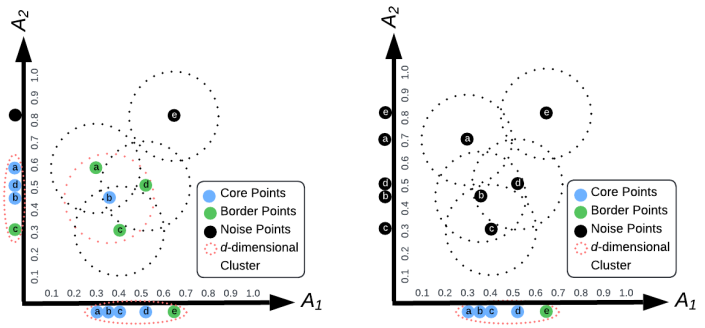
core points within their ε-neighborhood. A cluster is thus defined as the union of all density-connected sets, including both core and border points.

After detecting clusters in 1-dimensional subspaces, SUBCLU combines them into higher-dimensional subspaces using the following definition:

Definition 3. *Candidate k-dimensional subspaces are formed by $(k-1)$-dimensional subspaces that shares any $(k-2)$ dimensions.*

DBSCAN is then reapplied to these candidate subspaces using the same ε and *minPts* values to verify whether clusters from lower-dimensional subspaces persist as additional dimensions are considered. To limit the search space, SUBCLU prunes higher-dimensional subspaces based on Lemma 1.

Figure 3 illustrates an example where SUBCLU identifies a cluster with three points. The algorithm begins by classifying the points in each 1-dimensional subspace and then proceeds to higher-dimensional ones. Note from the figure, that the points may have different classifications accros dimension. For instance, point d is a core point in A_1 but a border point in A_2.



(a) One 2D cluster found.

(b) No 2D cluster found after a is moved further up along A_2.

Fig. 3. Illustration of SUBCLU.

3 Evaluation

The evaluation of the clustering algorithms was conducted on an Intel i7 1.70 GHz processor (12th gen.) with 16 GB of RAM, running Windows 11. MAFIA was evaluated using the *GPUMAFIA* implementation [1], which was installed on a virtual machine running Ubuntu, configured with 4 CPUs and 4 GB of RAM. CLIQUE and SUBCLU were evaluated on the main machine using *ELKI* [13]. Thus, one should be careful to compare the results of the three algorithms directly, as the execution environment and the implementation may affect the results. However, the growth rate and their clustering of the data sets can be compared.

In the experiments, a range of input parameters for the three algorithms was tested. The optimal parameters, based on the recommendations from [14,

p. 352], were selected to demonstrate the best performance for each dataset. The complete evaluation project can be found in the GitHub repository: https://github.com/henrikdchristensen/SDU-Data-Mining-Exam. Here, additional experiments as well as detailed descriptions of how to reproduce the experiments.

3.1 Data set generation

The goal of the synthetic data generation was to create similar datasets to those discussed in [2,12], featuring axis-parallel, hyper-rectangles, non-overlapping clusters in different subspaces. This was accomplished using *MDCGen* [7], which allows control over the size and density of clusters, as well as specifying which attributes act as noise for each cluster. Noise values were randomly selected from a uniform distribution across the full range of each attribute.

To highlight the main advantage of SUBCLU compared to CLIQUE and MAFIA, a dataset containing a Bezier-shaped cluster was generated using *Artificial Cluster* [11]. Additionally, a self-generated dataset with a plus-shaped cluster, as discussed in [12], was created for further evaluation.

Finally, two real-world data sets were selected to evaluate the algorithms in a more realistic setting.

All datasets were normalized so that each attribute is within the range [0, 1].

3.2 Experimental Results

Scalability with Data Set Size To evaluate the scalability in terms of the dataset size, a 20-dimensional dataset containing 5 clusters in 5 different subspaces with 10% noise records was used. The dataset sizes ranged from 10k to 1mio records. The results, shown in Figure 4, demonstrates that MAFIA is the most scalable algorithm of the three. CLIQUE could handle up to 500k records, while SUBCLU was only able to handle up to 100k records. Both CLIQUE and MAFIA exhibit linear growth rates, while SUBCLU shows a quadratic growth rate.

Note that, the *minPts* in SUBCLU were scaled linearly with the data set size, as the clusters in the dateset were of fixed sizes, which means, by increasing the data set size, the density of the clusters increases linearly. However, for the other two algorithms, there was no need to scale any of input parameters, as they relies on the density of the units for the total amount of records.

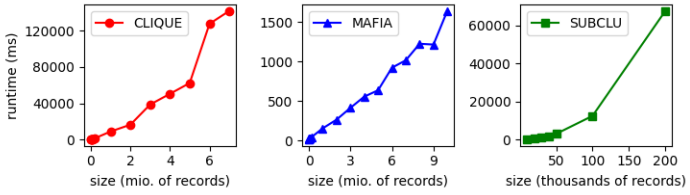


Fig. 4. Scalability with increasing data set size.

Scalability with Data Dimensionality and Cluster Dimensionality Figure 5 shows the scalability of the CLIQUE and MAFIA with increasing data set dimensionality. The data set contains 1mio. records with 3 clusters in 5 different subspaces, with 10% noise. The dimensionality ranges from 10 to 100 dimensions. However, CLIQUE could only handle half of the dimensions, as the PC freezes due to the high memory consumption. To investigate this further, one could try to reduce the memory consumption by having a smaller amount of records in the data set. Nevertheless, from the experiment both seems to scale linearly with the data set dimensionality.

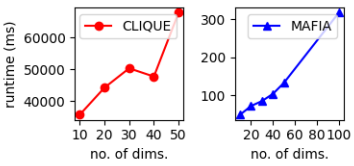


Fig. 5. Scalability with increasing data set dimensionality.

Figure 6 shows the scalability of the CLIQUE and MAFIA with increasing cluster dimensionality. A 20-dimensional data set, containing 500k records with a single cluster was used, with the cluster embedded in an increasing number of dimensions, ranging from 10 to 100 dimensions. Additionally, 10% of the records were noise. The data set is similar to the one used in [12]. CLIQUE exhibits quadratic growth, while MAFIA shows primarily linear growth. However, MAFIA's runtime suddenly increases once the number of dimensions exceeds 15.

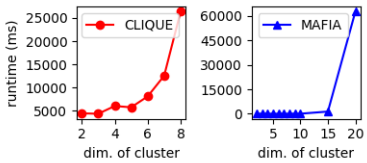


Fig. 6. Scalability with increasing cluster dimensionality.

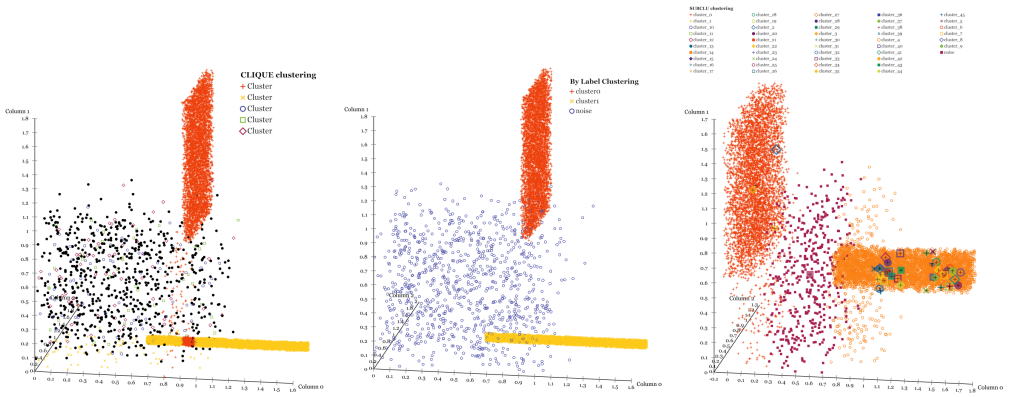
An attempt was made to replicate the results for SUBCLU reported in [10], but the PC freezes due to the high memory consumption. In future work, it would be valuable to explore this further, possibly using a smaller number of records in the dataset.

Clustering Accuracy Several synthetic datasets were generated to evaluate the clustering accuracy of the algorithms. It is important to note that the results were based on visual inspection of the clusters, as the external metrics available in ELKI are not well-suited for subspace clustering [5]. Although an attemt was made to implement the subspace clustering evaluation metric *E4SC* and the *clustering error* (CE) [5], the implementation was unsuccessful.

The first dataset was a 2-dimensional set with a single skewed-plus-sign shaped cluster and 10% noise. Results similar to those reported in [12] were

achieved. MAFIA successfully identified a single cluster that accurately captured the cluster's boundaries. In contrast, CLIQUE was unable to report a single cluster that correctly detected the borders. However, MAFIA's accuracy comes at a cost, as it also identifies many lower-dimensional clusters.

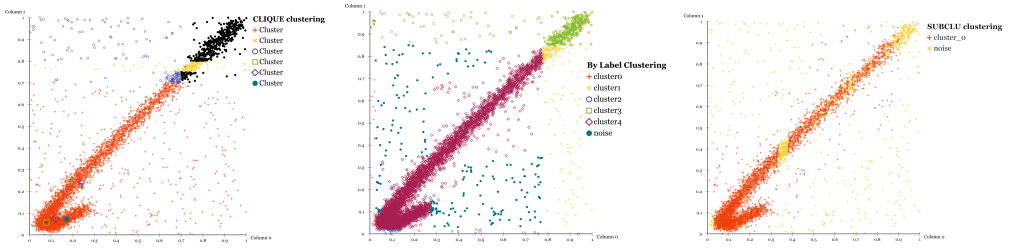
The second dataset was a 10-dimensional set containing 2 clusters embedded in a different 4 dimensional subspace, with 10% noise. This dataset was similar to the one used in [12]. The results are shown in Figure 7. MAFIA successfully detected both clusters without identifying any additional lower-dimensional clusters. In contrast, CLIQUE reports some overlapping clusters and misclassified some noise records as clusters. SUBCLU also detected the two clusters but, like CLIQUE, it identified many lower-dimensional clusters and classified noise records as clusters.



(a) CLIQUE **(b)** MAFIA **(c)** SUBCLU

Fig. 7. Two clusters in 4 different subspaces.

The third dataset were a 2-dimensional set containing a single cluster formed as a Bezier curve, with 10% noise. As expected, SUBCLU outperformed the other two, as shoen in Figure 8. MAFIA and CLIQUE only partially detected the cluster and misclassified additional clusters and noise records.



(a) CLIQUE **(b)** MAFIA **(c)** SUBCLU

Fig. 8. A single bezier-shaped cluster.

Real Data Sets Two real-world data sets are used to evaluate the algorithms in a more realistic setting. Small data sets were selected for both performance and visualization reasons. Both data sets were normalized to the range [0, 1].

The first data set is the well-known *Iris* data set [4], which contains 150 records of 3 different iris species with 4 features. All three algorithms successfully generated clusters, however, the overall clustering quality was suboptimal.

The second data set is the *Date Fruit* data set [8], consiting of 898 records of 7 different types of date fruit with 34 features. Only SUBCLU were able to produce clusters somewhat close to the true clusters but also finds many lower-dimensional clusters and small clusters. CLIQUE and MAFIA either produced too many clusters or close to one single cluster.

4 Discussion

Experiments have shown that MAFIA is the most scalable algorithm for large datasets due to its adaptive grid approach, which enhances both computational efficiency and clustering quality. While CLIQUE is also scalable, it struggles with clustering quality, especially when the grid size is not well-tuned. SUBCLU, on the other hand, excels in identifying arbitrarily shaped clusters through its density-connected approach, but this comes at a higher computational cost compared to MAFIA and CLIQUE.

Both CLIQUE and MAFIA have shown to scale linearly with the dataset size. In contrast, SUBCLU exhibits a quadratic growth, as it relies on partial range queries [10]. In terms of cluster dimensionality, CLIQUE exhibits a quadratic growth rate because the number of generated CDUs increases exponentially. In contrast, MAFIA generally shows a linear growth rate as cluster dimensionality increases. However, MAFIA's runtime rapidly increased once the number of dimensions exceeded 15, likely due to system limitations rather than the algorithm itself, as a linear growth rate is reported in [12]. Further research is needed to investigate this behavior. Due to high memory consumption, SUBCLU could not be tested on this dataset, but in [10] it is noted that it has at least a quadratic growth rate with respect to cluster dimensionality. Also the number of dimensions in the dataset does not affect growth rate of CLIQUE and MAFIA.

All three algorithms require careful parameter tuning for optimal performance, as observed in the clustering quality experiments. However, a key distinction between MAFIA and the other two algorithms is that MAFIA depends on both *cluster parameters* (i.e. α) and *algorithm parameters*, whereas CLIQUE and SUBCLU solely on algorithm parameters, such as the global density threshold [14, p. 342]. By using cluster parameters, MAFIA can set thresholds for how strong a cluster must be to be considered. However, this also complicates the tuning process by introducing more parameters to optimize. In addition to parameters like α and β, MAFIA also depends on the minimum number of bins (n) created at the start and the maximum number of windows (M) generated to prevent exponential growth as dimensionality increases. Both n and M were not extensively discussed in the original paper. The clustering accuracy experi-

ments clearly show that the performance of the algorithms is highly dependent on the structure of the data. It is also possible that better parameter combinations exist for an algorithm, which were not identified in these tests, and could further improve its performance. It should be noted, however, that algorithms less sensitive to parameter variations are generally preferred.

The trade-offs between scalability, clustering quality, and parameter sensitivity make it crucial to consider the specific needs of each application when choosing between CLIQUE, MAFIA, and SUBCLU. For large datasets where computational efficiency is a priority, MAFIA's adaptive grid approach offers significant advantages. However, for datasets with complex or irregularly shaped clusters, SUBCLU's density-based method is superior, albeit at a higher computational cost. As Kriegel et al. note [9, p.1:50], the trade-off between efficiency and effectiveness depends on the application.

5 Conclusion

In conclusion, this study highlighted the strengths and limitations of three subspace clustering algorithms. MAFIA's adaptive grid approach has proven to be the most scalable, making it ideal for large datasets where computational efficiency is required. Its flexibility in adapting to the data structure allows for better performance than CLIQUE, but this comes with the challenge of more complex parameter tuning. CLIQUE, while also scalable, suffers from reduced clustering quality in more complex data structures due to its reliance on fixed grid sizes. SUBCLU, with its density-connected method, excels at detecting arbitrarily shaped clusters, but its higher computational cost limits its scalability. MAFIA's ability to balance scalability and clustering quality, particularly in large-scale data sets, makes its a powerful algorithm in subspace clustering.

Future Work. In this study, internal and external evaluation metrics were not directly used to compare the clustering results of the three algorithms. The lack of external metrics is primarily due to the limitations of those available in ELKI, which are not well-suited for subspace clustering [5]. Therefore, it would be valuable to explore the subspace clustering evaluation metric *E4SC*, as proposed by [5]. Furthermore, a deeper investigation into MAFIA's sensitivity to parameter settings, as well as an exploration of SUBCLU's scalability with respect to cluster and data dimensionality, would be valuable.

References

1. Adinetz, A., Kraus, J., Meinke, J., Pleiter, D.: GPUMAFIA: Efficient Subspace Clustering with MAFIA on GPUs pp. 838–849 (08 2013). https://doi.org/10.1007/978-3-642-40047-6_83
2. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications p. 94–105 (1998). https://doi.org/10.1145/276304.276314, https://doi.org/10.1145/276304.276314
3. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. p. 226–231. KDD'96, AAAI Press (1996)
4. Fisher, R.A.: Iris. UCI Machine Learning Repository (1936). https://doi.org/https://doi.org/10.24432/C56C76
5. Günnemann, S., Färber, I., Müller, E., Assent, I., Seidl, T.: External evaluation measures for subspace clustering. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management. p. 1363–1372. CIKM '11, Association for Computing Machinery, New York, NY, USA (2011). https://doi.org/10.1145/2063576.2063774, https://doi.org/10.1145/2063576.2063774
6. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Elsevier (6 2011)
7. Iglesias Vázquez, F., Zseby, T., Ferreira, D., Zimek, A.: MDCGen: Multidimensional Dataset Generator for Clustering. Journal of Classification **36** (04 2019). https://doi.org/10.1007/s00357-019-9312-3
8. Koklu, M., Kursun, R., Taspinar, Y.S., Cinar, I.: Classification of Date Fruits into Genetic Varieties Using Image Analysis. Mathematical Problems in Engineering **2021**(1), 4793293 (2021). https://doi.org/https://doi.org/10.1155/2021/4793293, https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/4793293
9. Kriegel, H.P., Kröger, P., Zimek, A.: Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Trans. Knowl. Discov. Data **3**(1) (Mar 2009). https://doi.org/10.1145/1497577.1497578, https://doi.org/10.1145/1497577.1497578
10. Kröger, P., Kriegel, H.P., Kailing, K.: Density-Connected Subspace Clustering for High-Dimensional Data **246-257** (04 2004). https://doi.org/10.1137/1.9781611972740.23
11. Li, W., Zhou, Z.: AC: A data generator for evaluation of clustering (02 2022). https://doi.org/10.36227/techrxiv.19091330
12. Nagesh, H., Goil, S., Choud, A., Choudhary, P.: Adaptive Grids for Clustering Massive Data Sets (01 2002). https://doi.org/10.1137/1.9781611972719.7
13. Schubert, E.: Automatic indexing for similarity search in ELKI **13590**, 205–213 (2022). https://doi.org/10.1007/978-3-031-17849-8_16, https://doi.org/10.1007/978-3-031-17849-8_16
14. Sim, K., Gopalkrishnan, V., Zimek, A., Cong, G.: A survey on enhanced subspace clustering. Data Mining and Knowledge Discovery **26**(2), 332–397 (2 2012). https://doi.org/10.1007/s10618-012-0258-x, https://doi.org/10.1007/s10618-012-0258-x