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. Clustering is widely used in many applications, such as advertising and biology [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 a several challenges: First, the issue of *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. Secondly, the problem of *local 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 best fits the entire dataset. Also, algorithms that evaluate the entire feature space does not address this issue effectively. [9, p. 43–46]

Instead of relying on a global approach to feature selection, a local approach that addresses the second problem is necessary. However, then two separate problems needs to be dealt with, which both needs to be solved simultaneously. First of all, the clustering algorithm must be able to find the relevant subspaces for each cluster. Secondly, it must be able to find the clusters in each relevant subspace. To solve them efficiently, heuristics needs to be employed into the clustering algorithms. [9, p. 6–7]

For many applications, it is reasonable to focus only on clusters in axis-parallel subspaces, thus restricting the search space to $O(2^d)$ attributes. These algorithms are called *subspace clustering* (or *projected clustering*) algorithms, which uses one of two approaches: the *top-down* approach 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 rest of this paper, the following notation will be adopted: Let $\mathcal{A} = \{A_1, \ldots, A_d\}$ represent the set of attributes in a d-dimensional numerical space, $\mathcal{S} = A_1 \times A_2 \times \cdots \times A_d$, then a subspace is a subset of $\mathcal{A}$. Let $\mathcal{D}$ be the dataset, then 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

The primary focus of this paper is to analyze the grid-based bottom-up subspace clustering algorithm *MAFIA* [12], which builds upon the first grid-based subspace clustering approach *CLIQUE* [2]. The relationship between these two is examined. Additionally, the density-connected algorithm *SUBCLU* [10] is included, as it offers a contrasting approach to the grid-based approach. All three algorithms are also evaluated experimentally in terms of scalability and clustering quality.

The remainder of the paper is structured as follows: Section 2 describes and analyzes the three algorithms in detail. Section 3 evaluates their performance in terms of scalability, considering dataset size, data- and cluster-dimensionality, as well as their clustering quality. Section 4 discusses the findings and the methods used. Finally, Section 6 concludes the main findings and suggests future work.

2 Subspace Clustering

The overall idea of subspace clustering is to identify the subspaces that allows better clustering than the entire space $\mathcal{S}$. Two different approaches of subspace clustering will be discussed in this section including three subspace clustering

algorithms. All three is bottom-up approaches meaning they must implement the monotonicity property [9, p. 1:11], see Lemma 1.

2.1 Grid-Based approach

In a grid-based approach, $\mathcal{S}$ is partitioned into an axis-parallel grid structure. Each grid cell forms a axis-parallel, hyper-rectangular *unit*, and the number of points within each unit is counted. This process is initially applied in the 1-dimensional space for each attribute. Units that contain a number of points exceeding a predefined *density threshold* are retained as *dense units* (DUs). Then, these DUs are merged with adjacent ones in the next level (2-dimensional space) to form *candidate dense units* (CDUs), those of which are actually dense are kept. The final DUs are then the clusters found by the algorithm which will be described using minimal representations in the form of *Disjunctive Normal Form* (DNF) expressions using intervals of the attributes. Two algorithms that follows this approach are CLIQUE and MAFIA.

CLIQUE is the first proposed subspace clustering methods using a grid-based approach. It starts, by partitioning $\mathcal{S}$ into equal-sized *windows* (or *units*) of width ε (input parameter). Then, DUs are identified in for the 1-dimensional case, by counting the number of points in each window, for example, using a histogram, as shown in Figure 1, where each window has a certain *bin* count. For example, if $\varepsilon = 0.2$, then 3 DUs are identified in A_1, while 4 DUs are identified in A_2. The total of 7 DUs identified in the 1-dimensional space, will then be used to form CDUs in the 2-dimensional space. Specially, CLIQUE uses the following definition to form CDUs in 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 denote Definition 1 as the *first-approach*. Having formed the CDUs, then as mentioned, only the CDUs that are actually dense are kept for further processing.

To optimize performance, CLIQUE applies a pruning technique to the subspaces by calculating their *coverage*. Subspaces with low coverage (i.e., those containing few points) are pruned to reduce computation time. This approach helps manage the potentially large number of CDUs that CLIQUE may generate. However, this pruning strategy carries the risk of removing subspaces that might contain valuable cluster information.

Once the final DUs have been identified, CLIQUE seeks to group these DUs into clusters. A *region* is formed when adjacent DUs are connected 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 represent the extent of a cluster in the given subspace. For instance, in Figure 1, two distinct maximal regions, labeled A and B, are shown. The minimal cover for the cluster has the following DNF expression: $((0.2 \leq A_1 < 0.6) \wedge (0.4 \leq A_2 < 0.8)) \vee ((0.4 \leq A_1 < 0.8) \wedge (0.2 \leq A_2 < 0.6))$.

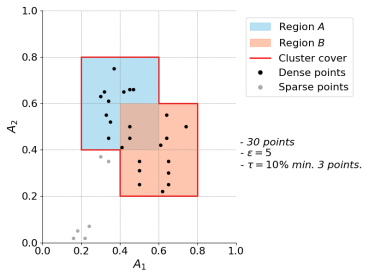


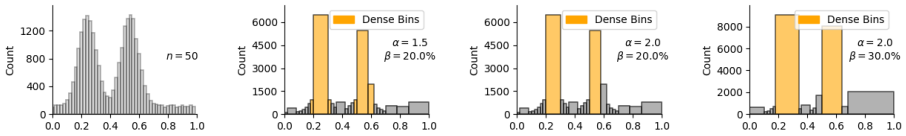
Fig. 1. 2-dimensional space containing 2 overlapping dense regions with 8 DUs in total.

MAFIA can be seen as an extension to CLIQUE and follows mostly the same procedure. However, MAFIA has the following changes:

1. Automatically determined variable sized grids based on the data distribution.
2. Forms CDUs using all combinations of the dimensions that the DUs contains.
3. Do not use the pruning technique as it could result in lost information, as already noted in [2].
4. Tries to parallelize the clustering process for higher performance.

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

The Adaptive Grid Algorithm starts by divide each attribute into high n-number of equal-sized windows (default $n = 1000$). Then, from left to right, two windows are merged together if their bin count are within a percentage of difference β (input parameter). That means, a high β value result in many merged windows, and vice-versa. If two windows are merged together, the highest bin count are assigned to the window, meaning the bin count that should be used to be compared with the next window. In other words, the algorithm does not use the total bin count of the two for the next comparison. Figure 2 shows an example before and after running the Adaptive algorithm using different values of α and β. Having merged the windows, the algorithm finally determines a *dense-level* for each of the windows, that determines how many points a window must contain for a unit to be dense, thus determines whether it should be a CDU for higher dimensional spaces. The dense-level is given by: 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* A_i covers. Thus, a higher α results in a higher dense-level, and vice-versa, as seen 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.

The CDU Generation of MAFIA is slightly different compared to CLIQUE. MAFIA uses the following definition when 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 denote Definition 2 as the *any-approach*. The difference between CLIQUE's *first-approach* and MAFIA's *any-approach* is that MAFIA explores any combination of dimensions that the DUs may share. Indeed, the *any-approach* could result in many combinations, and in fact the worst case running time is as high as $O(Ndu^2)$, where Ndu is the number of DUs. MAFIA can afford this high running time as it uses adaptive grids which reduces the number of generated DUs. Actually, CLIQUE also could implement the *any-approach*, however, big grid sizes may then be required to ensure few DUs are generated. Also note that, the *any-approach* can produce repeated CDUs, which therefore needs to be eliminated before proceeding to higher dimensions. Table 1 illustrates the advantage of the *any-approach*. Here, 3 DUs are provided to the a 4-dimensional space. Both CLIQUE and MAFIA would form a 4-dimensional CDU based on DU1 and DU2, as they share the *first* two dimensions. However, only MAFIA would form a CDU based on 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 of CDU generation of CLIQUE and MAFIA. The resulting CDUs are used in the 4-dimensional space.

2.2 Density-Connected approach

A drawback of grid-based methods is that the quality of clustering depends on the positiong of the grids, as seen in Figure 1. Moreover, clusters that do not conform to a hyper-rectangular shape are difficult to detect accurately. One way to address this issue is to use a density-connected algorithm like SUBCLU.

SUBCLU extends the principles of the well-known DBSCAN algorithm [3] to higher dimensions. The algorithm begins by applying DBSCAN to each attribute separately in 1-dimensional subspaces. It identifies dense regions using two input parameters: ε which defines the radius of points to be considered neighbors, and $minPts$, the minimum number of points required for a point to qualify as a *core point*. If a point is not a core point but lies within ε of a core point, it is classified as a *border point*. Points that are neither core nor border points are treated as *noise*.

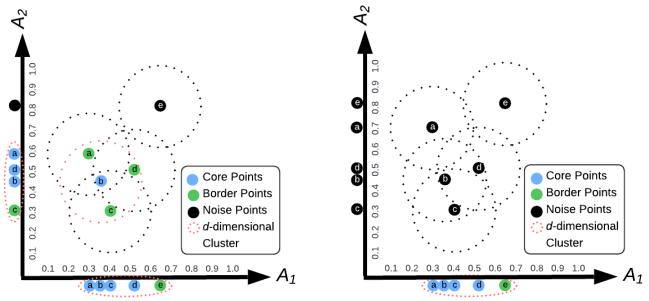
A *density-connected set* consists of core points that are linked through other core points, and border points that extend the cluster but do not meet the density criteria themselves. These border points are connected to the cluster via core points within their ε-neighborhood. A cluster is thus defined as the union of all density-connected sets, including both core and border points.

After detecting clusters in 1-dimensional subspaces, it uses the following definition to combine k-dimensional subspaces:

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

Note that, SUBCLU also uses the *any-approach* as MAFIA.

DBSCAN is reapplied to these candidate subspaces using the same ε and $minPts$ values to verify whether the clusters found in lower-dimensional subspaces persist as new dimensions are added. To reduce the search space, SUBCLU prunes higher-dimensional subspaces using Lemma 1. That is, if a cluster does not exist in a lower-dimensional subspace, that are pruned from consideration. An example is shown in Figure 3, where SUBCLU identifies a cluster containing three points. The algorithm begins by classifying the points in each 1-dimensional subspace and then proceeds to higher-dimensional subspaces. As seen, the points may not be classified the same way across the attributes in the k-dimensional space. For instance, point d is identified as a core point in subspace A_1, but is only a border point in subspace 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 performed on a Intel i7 1.70 GHz processor (12th gen.) with 16 GB of RAM running Windows 11. The evaluation of MAFIA was performed using *GPUMAFIA* [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.

Throughout the different experiments, a range of input parameters for the three algorithms were tested. The best found were selected to show their best performance on the corresponding data set. 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 aim for the synthetic data generation was to be able to produce similar data sets as discussed in both [2,12]. That is, axis-parallel, hyper-rectangles, non-overlapping clusters in different subspaces. This was achieved using *MDCGen* [7], where different sizes and different densities of the clusters could be set, as well as which attributes for each cluster that acts as noise for which values at random is selected from a uniform distribution over the entire range of the attribute.

However, to see the one of the main advantages of SUBCLU compared to CLIQUE and MAFIA, a data set containing a Bezier-shaped cluster was created using *Artificial Cluster* [11]. In addition, a self-populated data set containing a plus-shaped cluster was created as discussed in [12].

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

All data sets were normalized so that each attribute are in the range [0, 1].

3.2 Experimental Results

Scalability with Data Set Size To evaluate the scalability of the algorithms a 20-dimensional data set containing 5 clusters in 5 different subspaces with 10% noise records was used. The data set size ranges from 10k to 1mio records. The results can be seen in Figure 4. As can be seen, 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.

CLIQUE and MAFIA achieves a linear growth rate, and SUBCLU have a quadratic growth. The linear behavior of CLIQUE and MAFIA comes from the fact that the number of passes over the data set depends only on the cluster dimensionality. An increase in the size of the data set just means that more data has to be scanned on every pass over the data set while finding DUs resulting in a linear growth rate. SUBCLU, on the other hand, relies on DBSCAN which needs partial range queries, which are costly in terms of running time.

Similar growth rates is achieved for all algorithms as their original papers [12,2,10], however, the achieved running time in this experiment is better, which is probably due to better hardware or slightly different data sets.

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.

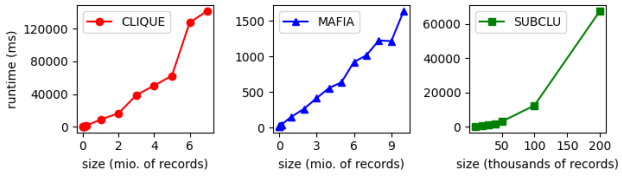
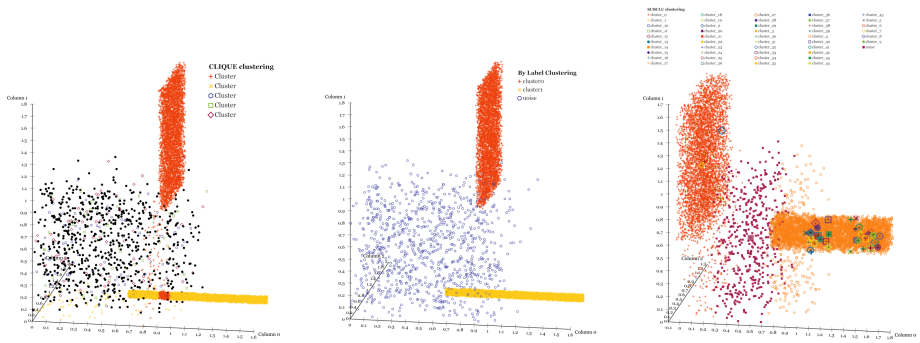


Fig. 4. Scalability with increasing data set size.

Clustering Accuracy Many different synthetic data sets were generated to test the clustering accuracy of the algorithms.

The first dataset was a 2-dimensional set containing a single cluster, shaped like a skewed plus-sign, with 10% noise added. 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.

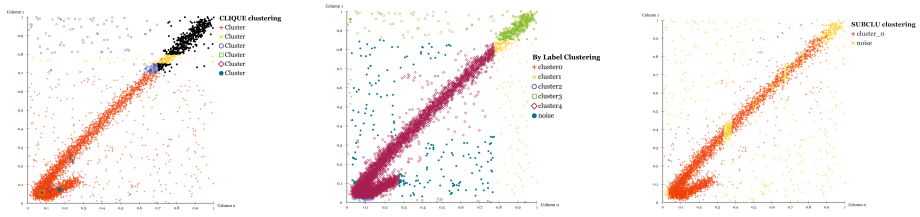
The second data was a 10-dimensional set containing 2 clusters embedded in a different 4 dimensional subspace. 10% of the data was added as noise records. Similar data set as the one used in [12]. The results can be seen in Figure 5. Here, MAFIA was able to detect the clusters without any additional lower-dimensional clusters. In contrast, CLIQUE reports some overlapping clusters and some of the noise records as clusters. SUBCLU detects also the two clusters, but reports many lower-dimensional clusters and noise records as clusters as well.



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

Fig. 5. Two clusters in 4 different subspaces.

The third data set were a 2-dimensional set containing a single cluster formed as a Bezier curve with 10% noise added to the data set. As expected, SUBCLU outperforms the other two algorithms as can be seen in Figure 6. MAFIA and CLIQUE only partly detects the cluster and reports additional clusters and noise records as clusters. The result of SUBCLU demonstrates its use of neighborhood of points.



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

Fig. 6. A single bezier-shaped cluster.

Scalability with Data Dimensionality and Cluster Dimensionality Figure 7 shows the scalability of the CLIQUE and MAFIA with increasing data set dimensionality. The data set contains 1 mio. records with 3 clusters in 5 different subspaces and 10% noise records. The data set dimensionality ranges from 10 to 100 dimensions. However, CLIQUE was only able to handle half of the dimensions, as the PC constantly freezes – probably due to the high memory consumption. In [2] it is noted that CLIQUE has a quadratic growth rate in terms of data set dimensionality. To investigate this even further, one could try to reduce the memory consumption by having a smaller amount of records in the data set. Nevertheless, the experiment clearly shows that MAFIA is the most scalable algorithm of the two.

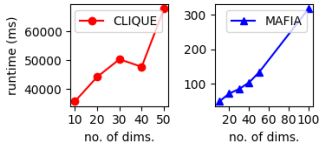


Fig. 7. Scalability with increasing data set dimensionality.

Figure 8 shows the scalability of the CLIQUE and MAFIA with increasing cluster dimensionality. A 20-dimensional data set, containing 500k records with a single cluster were used. The cluster was embedded in increasing number of dimensions starting from 10 to 100 dimensions. 10% of the records was added as noise. The data set is similar to the one used in [12]. Both algorithms heavily suffers from the increasing cluster dimensionality, however, MAFIA is able to handle a higher dimensional cluster than CLIQUE. The reason why both algorithms depends on the cluster dimensionality is due to the fact that a higher cluster dimensionality results in a large subspace coverage and a large number of CDUs. In other words, each pass on the data needs to populate a large number of CDUs, and increase in cluster dimensionality also increases the number of passes over the data set.

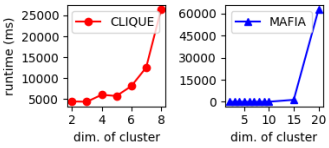


Fig. 8. Scalability with increasing cluster dimensionality.

The results reported for SUBCLU in [10] were also tried to be replicated, however, the PC constantly freezes – probably due to the high memory consumption. In future work, it would be interesting to investigate this further maybe using a very small data set.

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 such that each attribute was in the range [0, 1].

The first data set is the well-known *Iris* data set [4], which contains 150 records of 3 different iris types (e.g. Setosa, Versicolor, Virginica). The data set contains 4 features (sepal length, sepal width, petal length and petal width). All three algorithms successfully generated clusters using their corresponding approach. However, the overall cluster quality was not optimal.

The second data set is the *Date Fruit* data set [8], which contains 898 records of 7 different date fruit types (e.g. Barhee, Deglet Nour, Sukkary, Rotab). These were obtained via a computer vision, where 34 features (e.g shape and color) was extracted. 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 way too many clusters or close to one single cluster.

4 Discussion

Experiments have shown that MAFIA is the most scalable algorithm for large datasets, primarily due to its adaptive grid approach, which optimizes both computational efficiency and clustering quality. CLIQUE, while also scalable, struggles with clustering quality, particularly when the grid size is not well-tuned, limiting its ability to detect clusters in more complex data distributions. SUBCLU, on the other hand, excels at detecting arbitrarily shaped clusters through its density-connected approach, though this comes at a higher computational cost compared to MAFIA and CLIQUE. All three algorithms require careful parameter tuning for optimal performance, and they exhibit sensitivity to closely clustered data. The datasets used in the experiments can easily favor one algorithm over another depending on their structure.

One of the key differences between MAFIA and the other two algorithms is that MAFIA relies on both *cluster parameters* and *algorithm parameters*, whereas CLIQUE and SUBCLU use only algorithm parameters [14, p. 342]. Cluster parameters define the characteristics of the clusters themselves, while algorithm parameters guide the clustering process. This additional flexibility allows MAFIA to better adapt to the underlying data structure during clustering. However, this also complicates the tuning process as it introduces more parameters that must be optimized. Beyond parameters such as α and β, MAFIA also depends on the minimum number of bins (denoted n) and the maximum number of windows (denoted M). n controls the initial number of bins generated in the 1-dimensional space, while M limits the number of windows that can be generated to avoid exponential growth as the dimensionality increases. The choice of n and M is crucial for MAFIA's performance, but these parameters were not thoroughly discussed in the original paper.

A shared challenge across all three algorithms is their reliance on global parameters for determining which regions are considered dense. As noted in [9, p.1:16], using a global density threshold can lead to bias towards certain dimen-

sionalities, where a stricter threshold may miss high-dimensional clusters while a more relaxed one may generate too many low-dimensional clusters. MAFIA attempts to address this by adjusting the grid size adaptively, but the global parameter problem persists across all three methods.

The decision to use only the best parameters for the algorithms is based on the discussion in [14, p. 352], which highlights that such parameters are often "non-meaningful and non-intuitive." The authors recommend testing a range of parameters, iteratively adjusting them until an appropriate number of clusters is found. This same approach was applied in this study. However, it is possible that better parameter combinations exist, which were not identified in this process, and could further improve the performance of the algorithms. Nevertheless, we prefer algorithms which are not sensitive to different parameter values.

The trade-offs between scalability, clustering quality, and parameter sensitivity make it crucial to consider the specific requirements of each use case when choosing between CLIQUE, MAFIA, and SUBCLU. For large datasets where computational efficiency is key, MAFIA's adaptive grid approach offers significant advantages. However, for datasets containing complex or irregularly shaped clusters, SUBCLU's density-based method proves superior, albeit with a higher computational cost. Also, as noted by Kriegel et al. [9, p.1:50], "the trade-off between efficiency and effectiveness is tolerable will depend on the application".

5 Conclusion

In conclusion, this study highlighted the strengths and limitations of three subspace clustering algorithms, with a particular focus on MAFIA. MAFIA's adaptive grid approach has proven to be the most scalable, making it ideal for large datasets where computational efficiency is paramount. Its flexibility in adapting to the data structure allows for better performance, but this comes with the challenge of more complex parameter tuning. CLIQUE, while also scalable, suffers from reduced clustering quality in more complex data distributions 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, as the metrics available in ELKI are not well-suited for subspace clustering [5]. For future work, it would be valuable to investigate the subspace clustering evaluation metric *E4SC*, as proposed by [5]. Additionally, a deeper exploration of MAFIA's sensitivity to parameters could be further investigated.

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