Variable Density Based Clustering

Alexander Dockhorn, Christian Braune, and Rudolf Kruse
Institute of Intelligent Cooperating Systems
Department for Computer Science, Otto von Guericke University Magdeburg
Universitätsplatz 2, 39106 Magdeburg, Germany
Email: {alexander.dockhorn, christian.braune, rudolf.kruse}@ovgu.de

***Abstract*—The class of density-based clustering algorithms excels in detecting clusters of arbitrary shape. DBSCAN, the most common representative, has been demonstrated to be useful in a lot of applications. Still the algorithm suffers from two drawbacks, namely a non-trivial parameter estimation for a given dataset and the limitation to data sets with constant cluster density. The first was already addressed in our previous work, where we presented two hierarchical implementations of DBSCAN. In combination with a simple optimization procedure, those proofed to be useful in detecting appropriate parameter estimates based on an objective function. However, our algorithm was not capable of producing clusters of differing density. In this work we will use the hierarchical information to extract variable density clusters and nested cluster structures. Our evaluation shows that the clustering approach based on edge-lengths of the dendrogram or based on area estimates successfully detects clusters of arbitrary shape and density.**

I. Introduction

Extracting knowledge from data is the main goal of machine learning. In recent years databases showed a large increase in complexity, including a growing number of data set size and dimensions. Clustering describes the task of finding groups of similar objects. However, most clustering algorithms have limitations regarding shape and structure of clusters or finding appropriate parameter instantiations, restricting their application to modern data sets.

Despite the known drawbacks standard algorithms are often applied because of their simplicity. Nevertheless, non-expert users can have problems in interpreting results and adapting parameters to their use case. Many works focused on resolving known issues of specific clustering algorithms. In this work we will address limitations of the DBSCAN algorithm.

The DBSCAN algorithm proposed by Ester et al. [1], is the most commonly known representative of density-based clustering algorithms. Areas of higher density than the remainder of the data set are called clusters. Density is estimated as the number of objects (in this work also referred to as points) in a local environment. Fixing a radius for the environment and a minimal number of points in it, lets us detect areas of high density. This definition allows clusters of arbitrary shape. Nevertheless, estimating an appropriate threshold and size of the local environments is non-trivial.

In a previous work we proposed an alternating optimization procedure [2] to find an optimal parameter combination regarding a given objective function. The implementation was driven by two hierarchical adaptations of the DBSCAN algorithm, namely ε-HDBSCAN and m_{Pts}-HDBSCAN. In both of these the eponymous parameter was fixed and all valid clustering instantiations of the other parameter were computed. For an efficient computation of the hierarchies, we made use of the monotonicity of the parameter space. Regarding the parameter ε the hierarchy generation algorithm is similar to the one proposed by [3]. We adapted the procedure for the m_{Pts} parameter to create comparable hierarchies.

Our previous work was limited to horizontal cuts of the hierarchy. Those represent a DBSCAN clustering of a certain parameter combination. In this work we want to focus on elaborating on structural information about nested clusters, ultimately implementing non-horizontal cuts for detecting clusters of differing density. Section II will be used to reflect on the original DBSCAN algorithm and how to generate a hierarchy of clusters. In Section III we will present our approaches for non-horizontal cuts of the DBSCAN dendrograms. Our algorithms will be compared to local optimal DBSCAN clusters found by aoDBSCAN [2] in Section IV. Therefore, we test both algorithms on standard data sets and discuss their results. We will close this paper with a summary of the capabilities of non-horizontal cuts in Section VI.

II. Preliminaries

A. DBSCAN

This section will be used to recap the original DBSCAN algorithm [1]. It became well known for being one of the first density-based clustering algorithms.

The density of a local area is estimated by counting the number of elements in reach of a certain length. We will refer to the local surrounding of each point by the term ε-neighborhood, which is further defined by:

$$N_\varepsilon(p) = \{\, q \in D \,\middle|\, dist(p,q) \leq \varepsilon \,\} \tag{1}$$

The parameter ε describes the radius of a hypersphere centered at point p. All points with distance less or equal than ε will be contained in this hypersphere and, therefore, will be part of the ε-neighborhood. To filter for areas of high density we fix a minimum threshold for the size of the neighborhood sets. Each point, which size of the neighborhood set exceeds a value of m_{Pts}, will further be called core. The set of all core points can be described by the equation:

$$cores_{\varepsilon,m_{\text{Pts}}} = \{\, p \in D \,\middle|\, m_{\text{Pts}} \leq |N_\varepsilon(p)| \,\} \tag{2}$$

978-1-4799-7560-0/15/$31 ©2015 IEEE

The DBSCAN algorithm defines clusters as overlapping areas of high density. We will make use of the terms density-reachable and density-connected to further discriminate the relationship between two points. Let (directly) density-reachability be defined by:

Definition 1 ((directly) density-reachable): A point q is directly density-reachable from point p, if $q \in N_\varepsilon(p)$ and p is a core-point. Note that the conditions $p \in N_\varepsilon(q)$ and $q \in N_\varepsilon(p)$ are equivalent. Furthermore, two points $p,\ q$ are density-reachable if there exists a chain of points $p_1, \dots, p_n$ with $p_1 = p$ and $p_n = q$ such that for each $1 \leq i < n$, p_{i+1} is directly density-reachable from p_i.
Density-connectedness further loosens the restrictions of density-reachability by allowing two points to be connected if they have a common source of density-reachability.

Definition 2 (density-connected): Two points $p,\ q$ are density connected to each other if there exists a point o from which both points are density-reachable.

Finally, clusters are defined as the maximal set of points being density-connected to each other. Non-core points belonging to a cluster, therefore being density-reachable by at least one core point, are called border points. Points which neither fulfill the core-condition nor lie in the neighborhood of a core, will be determined as noise. Given a pair of ε and m_{Pts}, DBSCAN will automatically detect the number of clusters and the amount of noise in a data set. The average runtime of DBSCAN is $O(n \cdot log\ n)$, while the worst case complexity was shown to be $O(n^2)$ [4].

B. Creating Hierarchies of DBSCAN clusterings

While the original implementation of DBSCAN produces clusters of one specified density level, the monotonicity of the parameter space can easily be exploited to produce a hierarchy of clusterings. A basic observation is that increasing the radius of the neighborhood (ε) or decreasing the number of minimal points in each neighborhood (m_{Pts}) results in expanded clusters. Both observations will be formalized in the following.

Widening the radius of the neighborhood, done by increasing the value of ε, can result in points being added to the neighborhood set. This will always be the case when ε becomes greater than the distance of two points. In general we can show that for two radii $\varepsilon_1 > \varepsilon_2$ the following relationship holds true:

$$\begin{gathered} \{\, q \in D \,\big|\, dist(p,q) \leq \varepsilon_1 \,\} \supseteq \{\, q \in D \,\big|\, dist(p,q) \leq \varepsilon_2 \,\} \\ N_{\varepsilon_1}(p) \supseteq N_{\varepsilon_2}(p) \end{gathered} \tag{3}$$

The possible increase of the ε-neighborhood also influences the number of cores. For a fixed value of m_{Pts} we can infer:

$$\begin{gathered} \{\, p \in D \,\big|\, m_{\text{Pts}} \leq |N_{\varepsilon_1}(p)| \,\} \supseteq \{\, p \in D \,\big|\, m_{\text{Pts}} \leq |N_{\varepsilon_2}(p)| \,\} \\ cores_{\varepsilon_1, m_{\text{Pts}}} \supseteq cores_{\varepsilon_2, m_{\text{Pts}}} \end{gathered} \tag{4}$$

Similar effects can be achieved by relieving the core-condition, done by decreasing the value of m_{Pts}. This way further points can exceed the density threshold and will be added to the set of core points. For two density-thresholds $m_{\text{Pts}1} < m_{\text{Pts}2}$ the following relationship holds true:

$$\begin{gathered} \{\, p \in D \,\big|\, m_{\text{Pts}1} \leq |N_{\varepsilon}(p)| \,\} \supseteq \{\, p \in D \,\big|\, m_{\text{Pts}2} \leq |N_{\varepsilon}(p)| \,\} \\ cores_{\varepsilon, m_{\text{Pts}1}} \supseteq cores_{\varepsilon, m_{\text{Pts}2}} \end{gathered} \tag{5}$$

In both cases the number of core points needs to be updated after a change of the parameter. Starting with a small ε and increasing it stepwise until every point belongs to the same cluster will result in the production of a full hierarchy of clusters. Since each cluster can only grow in size, we can efficiently incorporate updates for each step by adding points which are now density reachable by a core point.

For this purpose, we define the minimal distance in which the size of the neighborhood set is greater or equal to the specified minimum number of points. We will further refer to this using the term core distance of a point and define it by:

Definition 3 (core distance): Let the core distance $d_{\text{core},m_{\text{Pts}}}(x_i)$ of a point $x_i \in \mathbf{X}$ be the distance to its m_{Pts}-nearest neighbor [5].

Using the core-distance we can define the distance in which a point is density-reachable to another. Therefore, we will use the term reachability distance and define it by:

Definition 4 ((mutual) reachability distance): Let the reachability distance of two points $x_i, x_j \in X$ be the distance, at which either x_i is density reachable by x_j or the other way around.

$$\begin{aligned} & d_{\text{reach},m_{\text{Pts}}}(x_i, x_j) \\ & \quad = \max\big\{ \min\{d_{\text{core, }m_{\text{Pts}}}(x_i),\ d_{\text{core, }m_{\text{Pts}}}(x_j)\},\ d(x_i, x_j)\big\} \end{aligned} \tag{6}$$

Symmetry can be forced by requiring both points to fulfill the core-condition. In this case we speak of mutual reachability distance.

$$\begin{aligned} & d_{\text{reach},m_{\text{Pts}}}(x_i, x_j) \\ & \quad = \max\big\{ \max\{d_{\text{core, }m_{\text{Pts}}}(x_i),\ d_{\text{core, }m_{\text{Pts}}}(x_j)\},\ d(x_i, x_j)\big\} \end{aligned} \tag{7}$$

Note that by forcing symmetry, as it was originally suggested by [3], border points would be ignored in the clustering process. We relaxed this condition to cover up the existence of border points, since they are characteristic of a stepwise merging process of two clusters.

Generating a hierarchy can trivially be done by iterating through a sorted list of reachability distances and updating neighborhood sets of involved clusters. Another approach adds the reachability distance of each pair of points $x_i,\ x_j$ as an edge with weight $d_{\text{reach},m_{\text{Pts}}}(x_i, x_j)$ between two homonym nodes to a graph. Computing a minimum spanning tree on this graph results in a condensed cluster hierarchy representation. The full hierarchy can be extracted by continuously removing the largest edge and noting the connected components of the graph. In case that the asymmetric reachability distance was used, only core-points can be used for determining such connected components. The original implementations of the hierarchy generation process can be reviewed in [2], [3].

III. Non-hierarchical cuts

The hierarchies produced in the previous section can be used in many ways. One is to fix an optimal horizontal cut height. This will result in clusters of about the same density. 1(a) and (b) show the result of this clustering approach using a dataset with a simple hierarchical structure. Nevertheless, the red, green and violet clusters are not completely the same and nested cluster structures get lost throughout the horizontal cut process. The complete hierarchical information is presented in Figure 1(c). The complete hierarchical information can be overwhelming for a fairly small data set. Our task will be to reduce the amount of nested structures, while still highlighting critical changes in density.

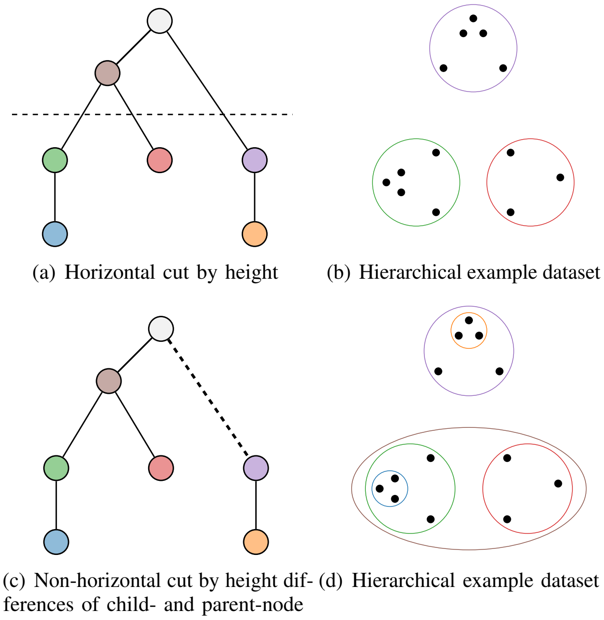
Taking a closer look at the example shows that the merging process between both clusters at the bottom needs less change in density than incorporating the cluster at the top. For an exact calculation of the density change we would need the number of points contained and the clusters area (or hyper-volume). The latter is a non-trivial task for arbitrary shaped clusters, since the calculation of a non-convex hull is complex and not well defined. We will provide two solutions for this process. In the first we will refrain from an area calculation and use the necessary parameter change of the merging process as an approximative density change of the cluster. In our second solution, we try to estimate the area of a 2D set of points by using alpha shapes and use this as our basis for a more profound density estimation. In Section IV we will compare achieved results of both algorithms.

A. Cut depending on parameter changes

Each edge in the dendrogram marks a transition in density of the cluster before and after the merge. Density separated structures will be merged by a larger edge than clusters with a smooth transition. For this reason, we will measure the height difference of clusters connected by an edge and retain subclusters (children node) in the final hierarchy if they were connected by a large edge. Sorting the edges by their height difference of children and parent node lets us fix a threshold for which edges will be cut. A valuable hyper-parameter in our observations was cutting edges longer than the 0.95 quantile of edge lengths. For our cut presented in Figure 1(c) this will result in one cluster at the top being a nested cluster of the complete data set. The groups of points at the bottom will just belong to the complete cluster since their density transitions are much more smoothly.

B. Cut depending on area of non-convex shapes

While our first method provides a fairly simple solution for density estimation, it does not account for the change of points between two merges. Adding just one point to the cluster would be weighted equally to merging two large clusters. Our second solution will make use of shape descriptors for clusters of arbitrary shape.



(a) Horizontal cut by height

(b) Hierarchical example dataset

(c) Non-horizontal cut by height differences of child- and parent-node

(d) Hierarchical example dataset

Fig. 1: Visualization of dendrogram cuts and the respective colorization of the data set

Algorithm 1 Edge Quantile Cut

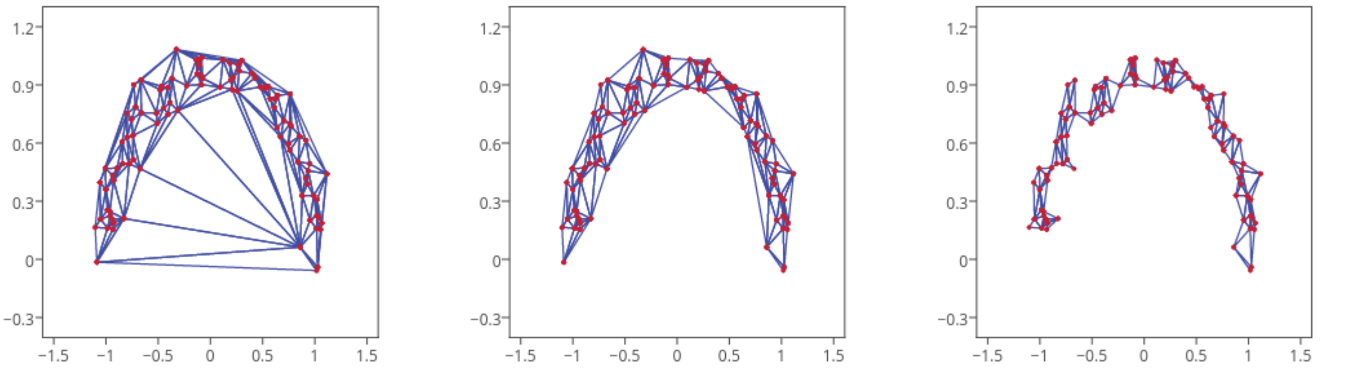
```
Input: $G$ = DBSCAN hierarchy, quantile

for all $(u, v) \in G.edges()$ from 1 to $N$ do
    height_before ← G.nodes(u).height)
    height_after  ← G.nodes(v).height)
    height_change ← height_after − height_before
    cutlist.append( (height_change,
                     G.nodes(u).points,
                     G.nodes(u).height))
end for

cutlist ← get_biggest_heightchanges(quantile)
for all $p \in$ Points do
    set $labels[p]$ to the index of the first
    cutlist element it is part of
end for
return labels
```

Calculating the area of a set of points can be solved by determining the convex hull throughout calculating the outer border of the Delaunay triangulation. See Figure 2(a) for an exemplary demonstration of simple data set and its Delaunay triangulation. Under the consideration of the density distribution, this convex hull can span sparse areas. We want to reduce the area spanned, while maintaining the characteristic shape of the data set.

To create a more appropriate non-convex hull of a set of points, we will make use of α-shapes proposed by Edelsbrunner et al. [6]. The α-complex of S is a subcomplex of the triangulation of S, which only contains α-exposed k-simplices $(0 \leq k \leq d)$. A simplice is α exposed if there is an open disc of radius $\sqrt{\alpha}$ through the vertices that do not contain any other



(a) Delaunay triangulation

(b) α-shape, $\alpha = 0.5$

(c) α-shape, $\alpha = 0.1$

Fig. 2: Delaunay triangulation and α-complex/shape for the moons data set

point of S, using the same metric as for the computation the underlying triangulation. The corresponding α-shape is defined as the interior space of the α-complex [7]. Two examples of α-shape are shown in Figure 2(b,c).

Algorithm 2 Alpha Shape Cut

```
Input: $G$ = DBSCAN hierarchy, quantile

cutlist ← list()
for all $(u, v) \in G.edges()$ from 1 to $N$ do
    area_before ← area_of (G.nodes(u).points)
    area_after  ← area_of (G.nodes(v).points)
    density_change ← area_after − area_before
    cutlist.append( (density_change,
                     G.nodes(u).points,
                     G.nodes(u).height))
end for

cutlist ← get_biggest_densitychanges(cutlist, quantile)
for all $p \in$ Points do
    set $labels[p]$ to the index of the first
    cutlist element it is part of
end for
return labels
```

IV. Experimental Setup

We evaluated our algorithms on multiple standard data sets available at [8]. The algorithm can be downloaded at [9]. Those clustering settings differ in size, density-distributions and incorporate flat and hierarchical structures. Our results are compared to performance values of our previous paper on aoDBSCAN [2], for which we used the density based silhouette coefficient as optimization criterion. We calculated the external validation measures entropy, purity, f-measure and v-measure to validate the clustering results.

We included evaluation values for the standard algorithms (fuzzy) c-means, hierarchical clustering using single linkage, complete linkage, and wards minimum variance criterion, as well as the two density-based algorithms optics and clique.

C-means, its fuzzy adaptation and the hierarchical agglomerative clustering algorithms all searched for a partition with the same number of clusters as the true cluster number of each dataset. Our evaluation of the OPTICS algorithm is based on the parameter settings $m_{\text{Pts}} = 4$ and 20 values of ε equidistantly distributed on a range of $[0.1, 1.00]$. Before applying the OPTICS algorithm the dataset was rescaled to the range of $[0.1, 1.00]$ to match the iterated ε-values. The result with the best density based silhouette coefficient was reported. The CLIQUE algorithm was initialized several times using gridsizes of (1010), (1515), (2020) and (2525). Results reported here are based on a density threshold of 4, which averaged as the best threshold value. The clustering with the best modified silhouette coefficient is recorded in the following evaluations.

Entropy is a concept of information theory, where it is used to describe the information contained in a message received. A clustering C can be seen as information source about the true structure P of the dataset and can be tested as predictor for exactly the same.

$$E(\mathcal{C}, \mathbf{P}) = -\sum_i p_i \left(\sum_j \frac{p_{ij}}{p_i} \log \frac{p_{ij}}{p_i} \right) \tag{8}$$

Entropy has a range of $[0, l \log K']$. Values near 0 describe an approximately perfect clustering, where each cluster C_i is congruent with a partition P_j. The maximal value of $\log K'$ states that any point in a cluster C_i is equiprobable to be in any partition P_j.

A closely related concept to entropy is purity, which is described by the extent in which a cluster contains objects of a single class [10].

$$P(\mathcal{C}, \mathbf{P}) = \sum_i p_i \left(\max_j \frac{p_{ij}}{p_i} \right) \tag{9}$$

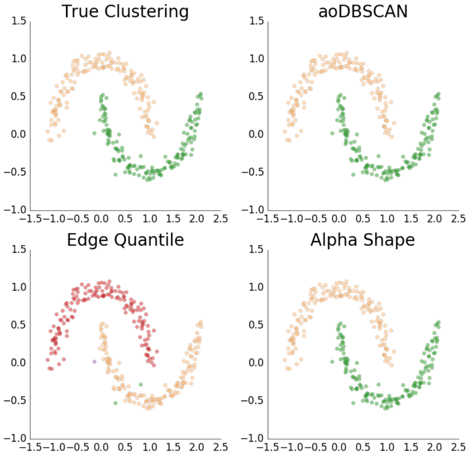


Fig. 3: Results on the Moons data set

TABLE I: Evaluation Moons data set

method	entropy	purity	v-measure	f-measure
aoDBSCAN	**0.00**	**1.00**	**1.00**	**1.00**
edge-quantile	0.09	0.99	0.89	0.96
α-shape	**0.00**	**1.00**	**1.00**	**1.00**
Mini-Batch-c-Means	0.81	0.75	0.75	*0.19*
Fuzzy-c-Means	0.81	0.75	0.75	*0.19*
Single Linkage	**0.00**	**1.00**	**1.00**	**1.00**
Complete Linkage	0.77	0.76	0.76	0.22
Wards Minimum Variance	0.38	0.89	0.89	0.60
OPTICS	0.49	0.80	0.81	0.42
CLIQUE	*1.02*	*0.67*	*0.70*	0.58

The range for the purity measure is $(0, 1]$. An optimal clustering assigns all nodes of a cluster to the same partition and therefor achieves a value of 1.

The f-measure is based on the harmonic mean of the two concepts precision and recall from the information retrieval community. The combination of both averages the extent in which clusters contain only objects of one true partition (precision) and the extent in which a cluster contains all elements of this partition (recall). The measure can be described by the following formula:

$$F(\mathcal{C}, \mathbf{P}) = \sum_j p_j \max_i \left(\frac{2 \frac{p_{ij}}{p_i} \frac{p_{ij}}{p_j}}{\frac{p_{ij}}{p_i} + \frac{p_{ij}}{p_j}} \right) \tag{10}$$

As it is the case for purity, the f-measure shares the range of $(0, 1]$, where an optimal clustering achieves a value of 1.

The v-measure is based on the harmonic mean of homogeneity and completeness. This combination follows a similar concept as the f-measure, but showed superior results in experiments by [11]. The v-measure can be calculated by:

$$V(\mathcal{C}, \mathbf{P}) = \frac{2 \cdot \mathit{Hom}(\mathcal{C}, \mathbf{P}) \cdot \mathit{Compl}(\mathcal{C}, \mathbf{P})}{\mathit{Hom}(\mathcal{C}, \mathbf{P}) + \mathit{Compl}(\mathcal{C}, \mathbf{P})} \tag{11}$$

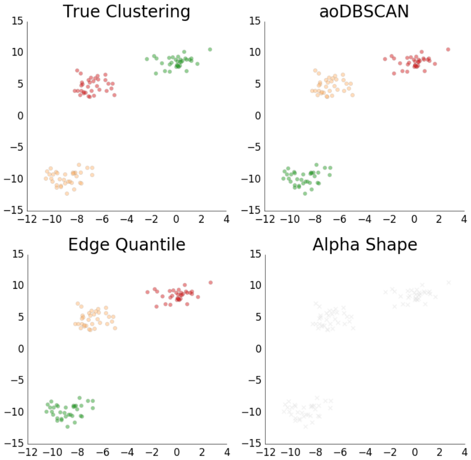


Fig. 4: Results on the Blobs-1000D showing first 2 dimensions

TABLE II: Evaluation Blobs-1000D data set

method	entropy	purity	v-measure	f-measure
aoDBSCAN	**0.03**	**1.00**	**1.00**	**1.00**
edge-quantile	**0.03**	**1.00**	**1.00**	**1.00**
α-shape	—	—	—	—
Mini-Batch-c-Means	**0.03**	**1.00**	**1.00**	**1.00**
Fuzzy-c-Means	**0.03**	**1.00**	**1.00**	**1.00**
Single Linkage	**0.03**	**1.00**	**1.00**	**1.00**
Complete Linkage	**0.03**	**1.00**	**1.00**	**1.00**
Wards Minimum Variance	**0.03**	**1.00**	**1.00**	**1.00**
OPTICS	**0.03**	**1.00**	*0.51*	*0.00*
CLIQUE	*0.17*	*0.97*	0.74	0.60

V. Results

The moons data set consists of two entwined sickles of nearly constant density. Figure 3 shows the results of the clustering process using an initial m_{Pts} values of 5. As it can be seen both non-horizontal cuts detect the clusters correctly. Additionally, the edge-quantile cut excludes some points in the lower sickle, since the ε-value has to be increased to include all those points in the cluster. The α-shape cut does not make these exclusions since the increase of the clusters area is negligible using an α-shape estimate. Since the density is constant in this data set aoDBSCAN is successful in determining an appropriate parameter combination.

In a next step we wanted to evaluate our algorithms in high dimensional space. For this purpose, we created a data set including three separated hyper-spheres in 1000 dimensional space. Finding appropriate configurations for ε becomes very hard if done by the user. aoDBSCAN finds appropriate ε and m_{Pts} values to distinguish all three clusters. The edge quantile is also able to find the clusters, because of the large increase of ε to merge the clusters. We currently cannot provide test results for the α-shape cut, since our implementation misses an hyper-volume estimate for polygon meshes.

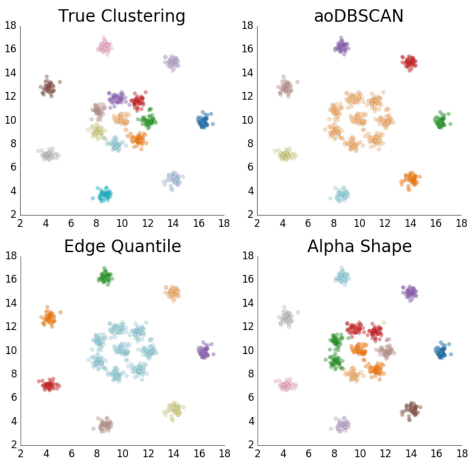


Fig. 5: Results on the R15 data set

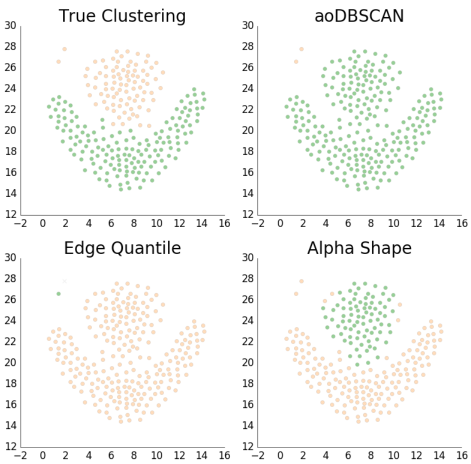


Fig. 6: Results on the Flame data set

TABLE III: Evaluation R15 data set

method	entropy	purity	v-measure	f-measure
aoDBSCAN	**0.00**	**1.00**	0.59	0.74
edge-quantile	**0.00**	**1.00**	0.59	0.74
α-shape	0.04	0.99	0.86	0.94
Mini-Batch-c-Means	0.02	**1.00**	**1.00**	**0.99**
Fuzzy-c-Means	0.02	**1.00**	**1.00**	**0.99**
Single Linkage	0.08	0.99	0.76	0.88
Complete Linkage	0.06	0.99	0.99	0.98
Wards Minimum Variance	0.05	0.99	0.99	**0.99**
OPTICS	**0.00**	**1.00**	0.59	0.74
CLIQUE	*0.25*	*0.94*	*0.43*	*0.58*

TABLE IV: Evaluation Flame data set

method	entropy	purity	v-measure	f-measure
aoDBSCAN	0.46	**0.99**	0.78	*0.02*
edge-quantile	0.45	**0.99**	0.77	*0.02*
α-shape	0.31	0.96	**0.96**	**0.77**
Mini-Batch-c-Means	0.02	0.85	0.85	0.48
Fuzzy-c-Means	0.05	0.85	0.85	0.44
Single Linkage	0.46	**0.99**	0.78	*0.02*
Complete Linkage	**0.00**	0.86	0.63	0.07
Wards Minimum Variance	0.11	0.72	0.72	0.33
OPTICS	0.46	**0.99**	0.78	*0.02*
CLIQUE	*0.80*	*0.52*	*0.55*	0.38

The R15 data set contains 15 clusters of constant density in which eight clusters in the middle form a dense and partly overlapping group. Those can only be separated by DBSCAN using a higher density threshold. Therefore, aoDBSCAN is not able do distinguish the clusters in the center. Edge-quantile cut recognizes the center as region of higher density, but does not separate contained clusters. The smooth transition between the clusters implies only slight changes in ε values per merge. In contrast α-shape cut is able to recognize merges between clusters, since the area doubles during a merge.

The flame data set highlights a smooth density transition between two clusters under the presence of two noise points. These add an extreme density transition into the hierarchy. aoDBSCAN was not able to detect an appropriate parameter combination using the density-based silhouette coefficient criterion, excluding only few points at the outer edges of the clusters. Because of the large area change at the moment of the merge, the α-shape cut is able to detect the merge between both clusters. The edge-quantile cut is not able to do the same, since the edge length distribution is skewed through the presence of noise.

After testing for smooth density transitions we incorporated a data set containing one large density change. The moons differing density data set consists of two sickles with large density variation. aoDBSCAN is not able to find a parameter combination yielding the desired clustering. This highlights the general incapability of aoDBSCAN to discriminate clusters with major differences in density levels. Both non-horizontal cuts are able to detect the changes, nevertheless, the huge amount of noise in the upper sickle prevents them from returning a perfect clustering result.

After testing for smooth density transitions we incorporated a data set containing one large density change. The moons differing density data set consists of two sickles with large density variation. aoDBSCAN is not able to find a parameter combination yielding the desired clustering. This highlights the general incapability of aoDBSCAN to discriminate clusters with major differences in density levels. Both non-horizontal cuts are able to detect the changes, nevertheless, the huge amount of noise in the upper sickle prevents them from returning a perfect clustering result.

We used the compound data set to test our algorithm's performance on a variety of hierarchical structures. While the two clusters in the bottom left corner form a group of constant density, both other groups contain clusters of differing density. aoDBSCAN is not able to find appropriate cluster parameters

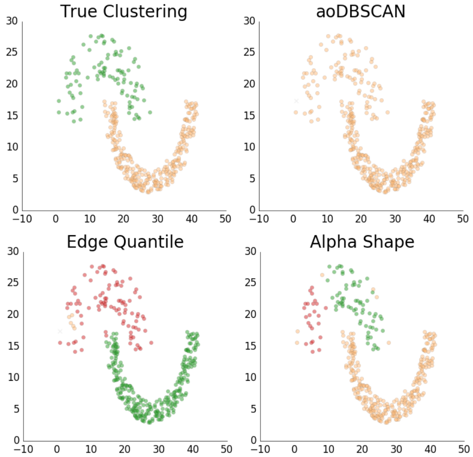


Fig. 7: Results on the Moons differing density data set

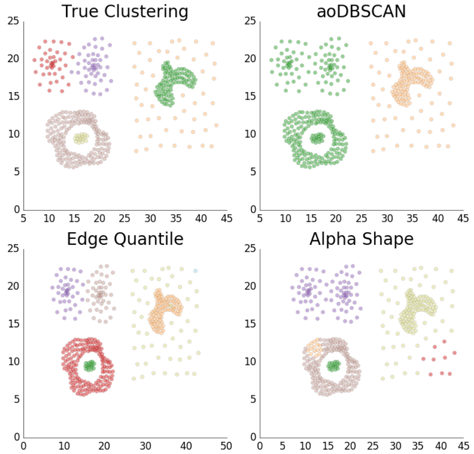


Fig. 8: Results on the Compound data set

TABLE V: Evaluation Moons differing density data set

method	entropy	purity	v-measure	f-measure
aoDBSCAN	0.41	0.99	0.85	0.01
edge-quantile	0.53	0.98	0.97	0.89
α-shape	0.68	0.92	0.92	0.78
Mini-Batch-c-Means	*1.09*	*0.76*	*0.74*	0.34
Fuzzy-c-Means	1.07	0.77	0.76	0.35
Single Linkage	0.61	0.93	0.85	0.25
Complete Linkage	0.58	0.95	0.95	0.70
Wards Minimum Variance	0.91	0.86	0.85	0.51
OPTICS	**0.39**	**1.00**	**1.00**	**1.00**
CLIQUE	0.85	0.84	0.84	0.22

TABLE VI: Evaluation Compound data set

method	entropy	purity	v-measure	f-measure
aoDBSCAN	0.74	1.00	0.77	0.59
edge-quantile	0.68	0.99	**0.99**	**0.98**
α-shape	0.02	0.75	*0.68*	0.74
Mini-Batch-c-Means	0.05	0.68	0.72	0.72
Fuzzy-c-Means	0.08	*0.66*	0.70	*0.71*
Single Linkage	0.69	0.99	0.84	0.80
Complete Linkage	0.50	0.91	0.84	0.81
Wards Minimum Variance	**0.01**	0.70	0.71	0.73
OPTICS	*0.74*	**1.00**	0.84	0.81
CLIQUE	0.45	0.93	0.88	0.77

under the presence of multiple reasonable density levels. Edge-quantile cut produced a nearly perfect result, miss-assigning just a few points in the transition of both clusters in the upper-left corner. The α-shape cut failed to return a correct clustering.

VI. Conclusions

In this paper we proposed two non-hierarchical cuts for DBSCAN dendrograms. Those utilize information about parameter and density changes to find nested clusters in hierarchical structured data sets, which condenses hierarchical information to meaningful clusters. Our algorithms performed well in various clustering scenarios, containing differing number of points, clusters and a variety of density changes.

Our first proposal, the edge-quantile cut, nearly perfectly clustered the compound data set, which consists of multiple nested cluster structures and density transitions. Nevertheless, same results could not be achieved in the flame data set, since the presence of noise seems to have a negative influence on this algorithm. The second algorithm, α-shape cut, is based on non-convex hulls. We estimate the hulls area of the points before and after the merge. Merges with a large increase in density highlight nested structures and will be kept. In our experiments α-shape cut excelled in identifying smooth density transitions between clusters, as it can be seen in the results for the R15 and the flame dataset. Produced α-shapes showed to be a good approximation of the non-convex shape. Nevertheless, it depends on the choice of an appropriate α value. Otherwise the area of the polygon diminishes too fast and it becomes hard to differentiate between clusters. We hope to improve on this using recently proposed shape descriptors by Braune et al. [12]. Another problem of our α-shape implementation is the current limitation to 2D-data sets. We would like to include hyper-volume estimation in a future work. An alternative would be to estimate the area size of each subspace.

In contrast to horizontal-cuts or DBSCAN parameter estimation algorithms, our hierarchy simplification algorithms are able to detect clusters of differing density. Hereby, expanding the set of use-cases for density-based clustering algorithms.

References

[1] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” *Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.

[2] A. Dockhorn, C. Braune, and R. Kruse, “An Alternating Optimization Approach based on Hierarchical Adaptations of DBSCAN,” in *2015 IEEE Symposium Series on Computational Intelligence (SSCI)*, no. 2, 2015, pp. 749 – 755.

[3] R. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," *Advances in Knowledge Discovery and Data Mining*, pp. 160–172, 2013.

[4] T. Ali, S. Asghar, and N. A. Sajid, "Critical analysis of DBSCAN variations," *2010 International Conference on Information and Emerging Technologies, ICIET 2010*, pp. 1–6, 2010.

[5] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure," *ACM SIGMOD Record*, vol. 28, no. 2, pp. 49–60, 1999.

[6] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transactions on Information Theory*, vol. 29, no. 4, pp. 551–559, jul 1983.

[7] H. Edelsbrunner and E. P. Mücke, "Three-dimensional alpha shapes," *ACM Transactions on Graphics*, vol. 13, no. 1, pp. 43–72, jan 1994.

[8] School of Computing, "Clustering datasets." [Online]. Available: https://cs.joensuu.fi/sipu/datasets/

[9] A. Dockhorn, "ao- and hierarchical db-scan." [Online]. Available: http://fuzzy.cs.uni-magdeburg.de/wiki-/pmwiki.php?n=Mitarbeiter.Dockhorn

[10] P.-N. Tan, M. Steinbach, and V. Kumar, "Chapter 8: Cluster Analysis: Basic Concepts and Algorithms," *Introduction to Data Mining*, pp. 487–586, 2005.

[11] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," *Computational Linguistics*, vol. 1, no. June, pp. 410–420, 2007.

[12] C. Braune, M. Dankel, and R. Kruse, "Obtaining Shape Descriptors from a Concave Hull-Based Clustering Algorithm," in *Advanes in Intelligent Data Analysis XV*. Springer International Publishing, 2016, pp. 61–72. [Online]. Available: http://link.springer.com/10.1007/978-3-319-46349-0_6