An Alternating Optimization Approach based on Hierarchical Adaptations of DBSCAN

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

***Abstract*—DBSCAN is one of the most common density-based clustering algorithms. While multiple works tried to present an appropriate estimate for needed parameters we propose an alternating optimization algorithm, which finds a locally optimal parameter combination. The algorithm is based on the combination of two hierarchical versions of DBSCAN, which can be generated by fixing one parameter and iterating through possible values of the second parameter. Due to monotonicity of the neighborhood sets and the core-condition, successive levels of the hierarchy can efficiently be computed. An local optimal parameter combination can be determined using internal cluster validation measures. In this work we are comparing the measures edge-correlation and silhouette coefficient. For the latter we propose a density-based interpretation and show a respective computational efficient estimate to detect non-convex clusters produced by DBSCAN. Our results show, that the algorithm can automatically detect a good DBSCAN clustering on a variety of cluster scenarios.**

I. Introduction

Clustering refers to the task of finding multiple sets of similar objects. While the related analysis task classification is in need of a class label for each instance, clustering algorithms find groups of similar items based on a similarity measure. Density-based clustering is a common type of clustering algorithms in which areas of higher densities, which are separated by low-density regions, are expected to form clusters. Instances located in sparse areas and not being part of a cluster are usually considered to be noise or outliers.

Prototype based algorithms such as c-means, fuzzy-c-means or expectation-maximization [1] expect the data set to include a given number of clusters c in which the instances have to be categorized into. This process often minimizes the pairwise dissimilarities between all objects in a cluster and maximizes inter-cluster-dissimilarities, which results in clusters of preferably convex shape.

In contrast density based methods such as DBSCAN [2] and OPTICS [3] are not limited to convex cluster shapes. Dense regions in any given shape can be detected as long as they fulfill the given density threshold. An additional benefit is that no estimation for the number of clusters has to be done. However, if available, it can be used to adjust the density threshold.

While density based methods perform very good on high-dimensional data sets [4], the estimation of a viable density level can become a challenging issue. As far as possible, multiple works have suggested estimates for appropriate parameter settings [5], [6] or reduce the number of free parameters (e.g. [7]).

This paper discusses the algorithm DBSCAN and a method for automatically determining locally optimal parameter settings given an internal cluster validation measure. Section II will give a quick overview of the algorithm DBSCAN and highlight the monotonicity of the eps-neighborhood and the core-condition with respect to the parameters eps and $minPts$. In Section III we propose two new algorithms based on these monotonicity observations. Furthermore we will explain how they exploit the monotonicity to create a hierarchy of clusterings by fixing one parameter and determining the optimal value for the other parameter. A local optimum of possible parameter combinations can be found using an alternating optimization approach. In addition, we compare available optimization criteria under the perspective of possible cluster structures produced by DBSCAN. Section IV contains multiple experiments on standard clustering data sets. We give a detailed view on the behavior of alternating optimization DBSCAN and compare the performance of optimization criteria considered.

II. Preliminaries

A. DBSCAN

In the year 1996, Ester et al. [2] proposed a density based clustering algorithm called DBSCAN, in which continuous regions of higher density are grouped in the same cluster. Therefore two parameters need to be fixed, the radius of a points neighborhood further referred to as eps and the minimal number of points $minPts$ for a region to be considered as dense region.

Let a region with a radius of eps centered at a point p of data set D be dense, if it contains at least $minPts$-points. For each point an eps-neighborhood can be defined as follows:

$$N_{\text{eps}}(p) = \{\, q \in D \,\big|\, dist(p,q) \leq eps \,\} \tag{1}$$

The eps-neighborhood equals the set of points inside a hypersphere with radius eps centered at p. A point is called core-point if its eps-neighborhood contains at least $minPts$ points.

$$cores_{\text{eps,minPts}} = \{\, p \in D \,\big|\, minPts \leq |N_{\text{eps}}(p)| \,\} \tag{2}$$

For a given pair of $(eps,\ minPts)$ a cluster is defined by the properties density-reachability and density-connectedness.

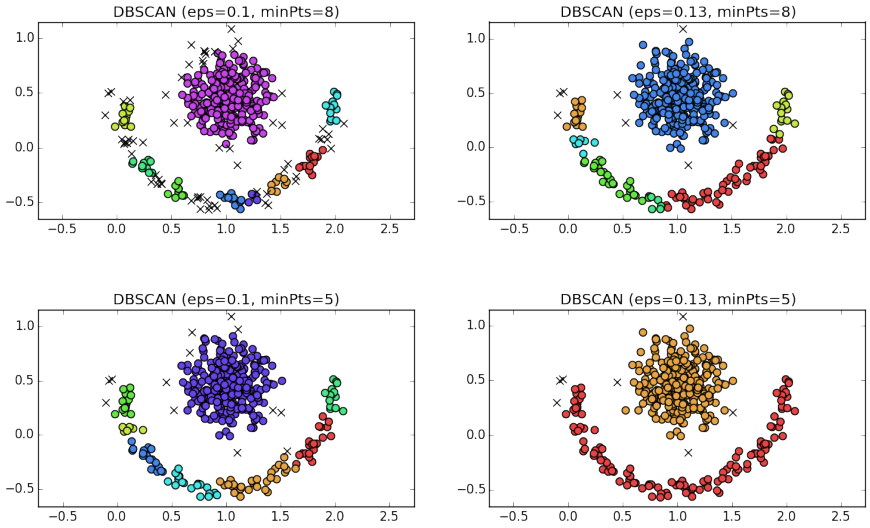

Fig. 1: Comparison of DBSCAN results for different parameter settings. Noise points are marked with '$\times$'. row-wise: monotonic behavior related to eps, column-wise: monotonic behavior related to $minPts$.

Definition 1 ((directly) density-reachable): A point q is directly density-reachable from point p, if $q \in N_{eps}(p)$ and p is a core-point. Note that the conditions $p \in N_{eps}(q)$ and $q \in N_{eps}(p)$ are equivalent. Furthermore, two points $p,\ q$ are density-reachable if there exists a chain of points $p_1, \ldots, p_n$ 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.

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.

A cluster C_i is a maximal set in which all points are density connected to each other. The set of all clusters is denoted as $\mathcal{C} = \{C_1, \cdots, C_k\}$. Note that a point, that is not a core point, can be part of more than one cluster, if it lies on directly on the border between two clusters. Points not located in any cluster can be considered noise.

For a given pair of eps and $minPts$, DBSCAN has an average runtime of $O(n \cdot log\ n)$, in which all clusters matching the density condition will be found.

B. Monotonicity of DBSCAN

Monotonicity is a frequently exploited property which enables algorithms to iterate through possible parameter values and adjust the current result to the new parameter. One typical example is the a-priori property in frequent item set mining [8], which states that the set of frequent item sets can not decrease if the minimum support is reduced. We will use a similar strategy by exploiting the monotonicity of the eps-neighborhood and the core-condition.

For two radii $eps_1 > eps_2$ we can show:

$$\begin{aligned} &|\{\, q \in D \,|\, dist(p,q) \leq eps_1 \,\}| \geq \\ &\qquad |\{\, q \in D \,|\, dist(p,q) \leq eps_2 \,\}| \\ &|N_{\text{eps}_1}(p)| \geq |N_{\text{eps}_2}(p)| \end{aligned} \tag{3}$$

The possible increase of the eps-neighborhood also influences the number of cores. For a fixed value of $minPts$ we can infer:

$$\begin{aligned} &\left|\{\, p \in D \,\middle|\, minPts \leq |N_{\text{eps}_1}(p)| \,\}\right| \geq \\ &\qquad \left|\{\, p \in D \,\middle|\, minPts \leq |N_{\text{eps}_2}(p)| \,\}\right| \\ &|cores_{\text{eps}_1,\text{minPts}}| \geq |cores_{\text{eps}_2,\text{minPts}}| \end{aligned} \tag{4}$$

Since the size of the neighborhood sets can only increase, it is possible that more points fulfill the core-condition. This can either increase the cluster sizes or merge multiple clusters into one, because additional points increase the density-reachability and density-connectedness. Figure 1 shows a comparison of multiple DBSCAN clusterings. Each row contains clusterings initialized with the same $minPts$ value, each column contains clusterings initialized with the same eps value. Changes in the clustering result can be attributed to a change of the neighborhood sets.

A similar approach can be used regarding the value of $minPts$. For two values $minPts_1 < minPts_2$ the following inequality holds:

$$\begin{aligned} &\left|\{\, p \in D \,\middle|\, minPts_1 \leq |N_{\text{eps}}(p)| \,\}\right| \geq \\ &\qquad \left|\{\, p \in D \,\middle|\, minPts_2 \leq |N_{\text{eps}}(p)| \,\}\right| \\ &|cores_{\text{eps},\text{minPts}_1}| \geq |cores_{\text{eps},\text{minPts}_2}| \end{aligned} \tag{5}$$

The neighborhood of each point is unaffected by the change of $minPts$. Note that a decrease of $minPts$ cannot decrease the size of the core-set. The columns of Figure 1 show DBSCAN clusterings for constant values of eps and differing $minPts$. Our proposed algorithms will now utilize the monotonicity observations for producing a hierarchy of clusterings.

III. HIERARCHICAL DBSCAN

We created two hierarchical variants of the DBSCAN algorithm based on the monotonicity criteria presented in Equations 4 and 5. For each variant, one of the two parameters has to be fixed in order to produce a hierarchy of clusterings by iterating through possible values for the second parameter. The following sections will take a closer look at both versions of our hierarchical DBSCAN algorithms.

A. minPtsDBSCAN

Relating to the core condition it is straight-forward to adjust the radius of a points neighborhood. For each point we determine a core-distance by randomly setting the parameter $minPts \leq |D|$ to a fixed value. Sorting the nodes by their core-distance gives us levels of the desired hierarchy. Consecutive levels include changes by either adding a new cluster, extending an existing cluster or merging two clusters which are now density-connected. Further hierarchy levels in between two core distances d_i, d_j might possibly exist, since new points can get in range of an existing core with distance d_k, where $d_i < d_k < d_j$.

From this observation follows that all pairwise distances of two points have to be processed to ensure that each different DBSCAN level is included in the hierarchy. The process can be stopped as soon as every point is a core-point and in the same cluster. The minPtsDBSCAN algorithm is summarized in the following pseudo-code:

Algorithm 1 minPtsDBSCAN

Input: min_pts, dist_mat = pairwise distance matrix of D

$\mathcal{C} \leftarrow \{\}$
clust_hierarchy $\leftarrow initialize\_clustertree(\mathcal{C})$
dist_list $\leftarrow$ sort dist_mat in priority list as $(r,\ c,\ d)$
(row-index, column-index, distance)

for all $(r,\ c,\ d)$ in dist_list **do**
  add c to neighborhood of r
  if $r \notin$ cores **and** $min\_pts \leq |N(r)|$ **then**
    add r to cores
  end if
  $update\_density\_reachability(r)$
  $\mathcal{C}_{dist} \leftarrow update\_clustering(\mathcal{C})$
  if $\mathcal{C}_{dist} \neq \mathcal{C}$ **then**
    clust_hierarchy.$add\_clustering(\mathcal{C}_{dist})$
    $\mathcal{C} \leftarrow \mathcal{C}_{dist}$
  end if
end for

return clust_hierarchy

For the algorithm minPtsDBSCAN we need to estimate a value for the parameter $minPts$. Choosing a value of $minPts = 1$ results in a hierarchical clustering with single-linkage. Note that changing the value of $minPts$ also changes the set of hierarchy levels. The functions $update\_density\_reachability$ and $update\_clustering$ can efficiently be obtained by reusing the status of the last computed distance.

B. epsDBSCAN

Our second algorithm is based on the stepwise decrease of $minPts$ while eps is constant. The monotonicity shown in Equation 5 implies that while decreasing the minimum number of points no cluster can decrease in size. Based on this observation, we propose another variant of DBSCAN called epsDBSCAN.

On initialization possible values for eps are in the range of $[min(dist\_mat),\ max(dist\_mat)]$. Our algorithm starts by setting $minPts$ equal to the biggest neighborhood-set size in the data set and decreasing it stepwise until every node fulfills the core-condition. In comparison to minPtsDBSCAN no further changes will be observed after each node became a core, since the neighborhood set is not influenced by the variable $minPts$. The set of hierarchy levels can be determined by:

$$\{\, |N_{eps}(p)| \,\big|\, p \in D \,\}$$

In minPtsDBSCAN the hierarchy levels were bound to pairwise distances, which results in a real number value, whereas epsDBSCAN is limited to natural numbers. The epsDBSCAN algorithm is summarized in the following pseudo-code:

Algorithm 2 epsDBSCAN

Input: eps, dist_mat = pairwise distance matrix of D

$\mathcal{C} \leftarrow \{\}$
clust_hierarchy $\leftarrow initialize\_clustertree(\mathcal{C})$
calculate neighborhood sets based on eps
minPts $\leftarrow max(|N_{eps}(p)|)$

while $|cores| \leq |D|$ **do**
  new_core_nodes = $\{\, p \in D \,\big|\, |N_{eps}(p)| =$ min_pts $\,\}$
  add new_core_nodes to cores
  $update\_density\_reachability$(new_core_nodes)
  $\mathcal{C}_{minPts} \leftarrow update\_clustering(\mathcal{C})$
  if $\mathcal{C}_{minPts} \neq \mathcal{C}$ **then**
    clust_hierarchy.$add\_clustering(C_{minPts})$
    $\mathcal{C} \leftarrow \mathcal{C}_{minPts}$
  end if
  $min\_pts \leftarrow min\_pts - 1$
end while

return clust_hierarchy

For epsDBSCAN it occurs far more often that nodes have to be added to the set of cores for the same value pair $(minPts, eps)$. Motivated by this observation, nodes are grouped by their neighborhood size in advance. This ensures a faster computation, since the methods for updating the density-reachability and the clustering are only called once per group of added cores.

C. Alternating Optimization

Both previous proposed algorithm can be combined to an alternating optimization process to find a locally optimal parameter combination. The produced cluster hierarchy given a fixed parameter combination can be analyzed in order to find an optimal value for the second parameter. This is done by rating the clustering of each hierarchy level given an internal cluster validation measure.

The alternating optimization process is summarized in the following pseudo-code:

Algorithm 3 aoDBSCAN

Input: dist_mat = pairwise distance matrix of D

min_pts ← random() or set by user
repeat
  clust_hierarchy ← minPtsDBSCAN (min_pts)
  eps = get_best_eps(clust_hierarchy)

  clust_hierarchy ← epsDBSCAN (eps)
  $minPts$ = get_best_minPts(clust_hierarchy)
until convergence of min_pts and eps

return min_pts, eps

It is not necessary to start the process with minPtsDBSCAN. In certain scenarios it can be beneficial to swap the order of minPtsDBSCAN and epsDBSCAN. Since the initial estimation influences the found local optima we recommend to use either epsDBSCAN or minPtsDBSCAN, depending on which parameter is easier to estimate. It proved beneficial to start with minPtsDBSCAN, since in average the number of hierarchy levels is smaller than for epsDBSCAN.

Given an optimization criterion we can rate each possible cut-height and choose the best in regard to the rating. In the following subsections we compare the measures silhouette score [9] and edge correlation [10] as two such optimization criteria.

1) (Density) Silhouette Coefficient: The silhouette coefficient is based on the tightness of a cluster in comparison to its separation to other clusters. A silhouette of point i is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{6}$$

where $a(i)$ is the average distance to points in the same cluster as point i and $b(i)$ the minimum distance to points of other clusters. The silhouette coefficient of a cluster is defined as the arithmetic mean of all silhouettes of points in a cluster C_i.

$$s_{C_i} = \frac{1}{n_{C_i}} \sum_{o \in C_i} s(o)$$

Accordingly the silhouette score is the mean of the silhouette coefficients of each cluster.

$$s_{\mathcal{C}} = \frac{1}{n_{\mathcal{C}}} \sum_{C_i \in \mathcal{C}} s_{C_i} \tag{7}$$

The inclusion of the average distance to each point in the cluster makes the silhouette score favor convex clusters. This is no problem for algorithms that tend to produce such shapes, but heavily limits the area of application for the DBSCAN algorithm. Therefore we propose a density-based interpretation of the silhouette score, which changes the definition of $a(i)$.

Let $G = (V, E)$ be a graph, where V is the set of all nodes of the cluster C_i. We start with $E = \emptyset$. For each pair of nodes $u, v \in V$ we can add an edge with weight $dist(u, v)$. All possible edges are sorted in ascending order by their edge weight and added to the graph until each pair of nodes u, v is connected by a path $p(u, v) = u \to v$. We define the cost of a path $cost(p)$ to be the largest edge weight on the path.

Let $a(i)$ be:

$$a(i) = \frac{1}{n_{C_i} - 1} \sum_{j \in C_i;\ i \neq j} \arg\min_p(cost(p(i, j)))$$

Since this formula is too complex to evaluate for every hierarchy level, we suggest to use the eps value, for which the cluster C_i first emerged, as an upper-bound $\bar{a}(i) \geq a(i)$.

$$\bar{a}(i) = \arg\min_{eps}(\text{for } C_i \text{ to exist})$$

Another problem that can emerge is the possible existence of border points belonging to more than one cluster. If we exclude all border points, the inequality $b(i) > \bar{a}(i)$ holds for any clustering produced by DBSCAN. Otherwise, there would exist a point p not present in $i \in C_i$, for which the distance $dist(p, i) = b(i) \leq eps$. In this case point p would be in the neighborhood of i and therefore the Cluster C_i would not be maximal.

The evaluation section will show that these changes in the calculation of $s_{\mathcal{C}}$ adapt well to cluster structures produced by DBSCAN. We will use the abbreviation $s_{\mathcal{C}}$ for results based on the original silhouette coefficient and $s^d_{\mathcal{C}}$ for the density based interpretation of silhouette coefficient, where $a(i)$ is replaced by $\bar{a}(i)$ and border points are excluded from the calculation.

2) Edge Correlation: Correlation clustering can be seen as maximizing the correlation of the clustering $\mathcal{C}$ and a given similarity measure. Based on an analysis of correlation clustering for graphs in [10], we calculate the Pearson-correlation ρ of a $n \times n$ cluster matrix $L_{\mathcal{C}}$ and a similarity matrix S.

Where the matrix L is defined by:

$$L_{\mathcal{C}}(i, j) = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are in the same cluster} \\ 0 & \text{else} \end{cases}$$

The entry (i, j) of $L_{\mathcal{C}}$ is 1 if i and j are in the same cluster referring to $\mathcal{C}$, and they 0 if they are not.

We used $1 - dist(i, j)$ as a similarity measure to form matrix S. Since we are only measuring the correlation between the similarity matrix S and the cluster matrix there is no need to use any special similarity measure. The only requirement is that the mapping from distance to similarity measure is linear. The edge correlation $\rho_{\mathcal{C}}$ of a clustering $\mathcal{C}$ can be calculated as seen in:

$$\rho_{\mathcal{C}} = \rho(L_{\mathcal{C}}, S)$$

TABLE I: Results for the Aggregation data set

measure	eps	minPts	iterations	homogeneity	completeness	v-measure
$s_{\mathcal{C}}$	1.423	5	2	0.80	0.99	0.89
$s_{\mathcal{C}}^{d}$	1.451	1	4	0.80	1.0	0.89
$\rho_{\mathcal{C}}$	3.482	11	4	0.78	1.0	0.84

TABLE II: Results for the Moons data set

measure	eps	minPts	iterations	homogeneity	completeness	v-measure
$s_{\mathcal{C}}$	0.224	5	2	1.0	0.52	0.68
$s_{\mathcal{C}}^{d}$	0.324	3	3	1.0	1.0	1.0
$\rho_{\mathcal{C}}$	0.210	1	4	1.0	0.40	0.57

TABLE III: Results for the Blobs-1000D data set

measure	eps	minPts	iterations	homogeneity	completeness	v-measure
$s_{\mathcal{C}}$	47.277	31	5	1.0	1.0	1.0
$s_{\mathcal{C}}^{d}$	141.315	5	2	1.0	1.0	1.0
$\rho_{\mathcal{C}}$	47.277	31	5	1.0	1.0	1.0

TABLE IV: Results for the Spirals data set

measure	eps	minPts	iterations	homogeneity	completeness	v-measure
$s_{\mathcal{C}}$	3.668	3	2	1.0	1.0	1.0
$s_{\mathcal{C}}^{d}$	1.107	1	4	1.0	1.0	1.0
$\rho_{\mathcal{C}}$	0.863	5	2	0.29	0.35	0.32

TABLE V: Results for the R15 data set

measure	eps	minPts	iterations	homogeneity	completeness	v-measure
$s_{\mathcal{C}}$	0.422	5	2	0.87	0.96	0.91
$s_{\mathcal{C}}^{d}$	0.625	1	3	0.59	1.0	0.74
$\rho_{\mathcal{C}}$	0.783	5	2	0.59	1.0	0.74

TABLE VI: Results for the D31 data set

measure	eps	minPts	iterations	homogeneity	completeness	v-measure
$s_{\mathcal{C}}$	1.025	10	5	0.82	0.92	0.87
$s_{\mathcal{C}}^{d}$	2.225	1	3	0.04	0.97	0.08
$\rho_{\mathcal{C}}$	1.163	2	3	0.53	0.98	0.69

TABLE VII: Results for the Flame data set

measure	eps	minPts	iterations	homogeneity	completeness	v-measure
$s_{\mathcal{C}}$	2.661	48	3	0.80	0.48	0.60
$s_{\mathcal{C}}^{d}$	1.254	1	4	0.01	0.18	0.02
$\rho_{\mathcal{C}}$	0.992	6	3	0.93	0.79	0.86

IV. Evaluation

We evaluated our proposed algorithm aoDBSCAN on a variety of data sets. The alternating optimization process was initialized using the optimization criteria silhouette coefficient $s_{\mathcal{C}}$, density based silhouette coefficient $s^d_{\mathcal{C}}$ and edge correlation $\rho_{\mathcal{C}}$. Each run started with an initial value of $minPts = 5$. This should be seen as a trade-off between single linkage behavior ($minPts = 1$) and focusing on areas of high density ($minPts >> 1$). Note that other local optima could be found using a different initial value for $minPts$ or starting with epsDBSCAN.

We used the external validation measures homogeneity, completeness and v-measure [11] to validate the clustering results. Homogeneity is highest if only data points of a single class were assigned to a single cluster. Symmetrically, completeness will be maximal if all data points of a single class were assigned to a single cluster. The weighted harmonic mean of homogeneity and completeness is called v-measure. Tables I to VII summarize the optimization process per optimization criterion.

The Aggregation data set contains clusters of various shape, where two groups, each containing two clusters, are connected by a bridge of lower density. All optimization criteria could adapt to the various cluster shapes, while silhouette coefficient and edge correlation reported few noise points and scored not as high as density based silhouette coefficient. However none of the optimization measures led to a separation of the connected clusters.

Data sets including non-convex clusters such as Moons and Spirals were best clustered using density based silhouette coefficient. Edge correlation performed worst on both scenarios, due to the high distances of points in the same cluster.

The performance on convex clusters was tested on the data sets Blobs-1000D, R15 and D31. While the first data set included 3 spherical clusters of 100 points each in 1000 dimensions per cluster, clusters in the data sets R15 and D31 consisted of less points with only 2 dimensions. The silhouette coefficient performed best in detecting these clusters. No effects of higher dimensionality were observed.

The data set Flame is made of a Gaussian distributed convex cluster and a cluster following a curved line around the first cluster. Both clusters are connected by a lower density area. Edge correlation was the only optimization criterion, which detected a nearly optimal clustering. The silhouette coefficient and its variant were unable to separate both areas at the lower density region.

V. Conclusions

In this paper we proposed two algorithms minPtsDBSCAN and epsDBSCAN, which are hierarchical versions of the widely known DBSCAN algorithm. By exploiting observed monotonies the algorithms can be efficiently implemented via actualizing previous clustering results.

Furthermore, the two algorithms can be used in an alternating optimization to find a local optimal parameter combination for DBSCAN. As a drawback an internal cluster validation

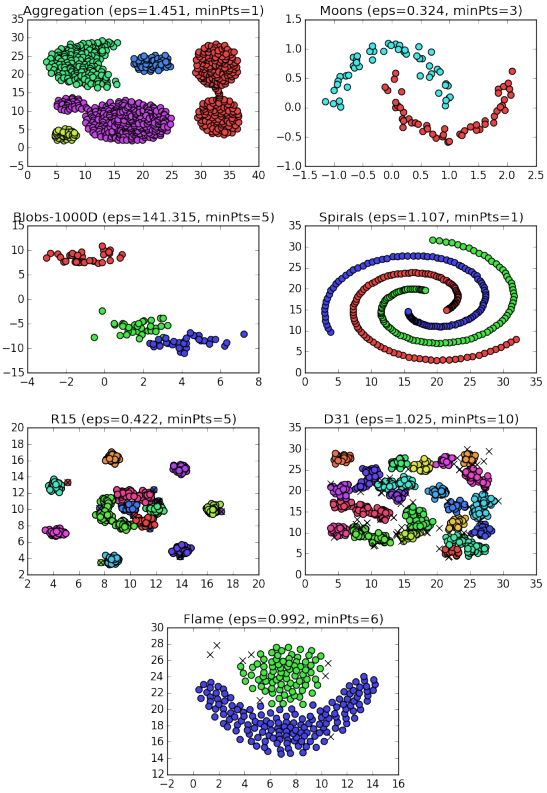

Fig. 2: Best clustering results for aoDBSCAN and an initial value of $minPts = 5$. The clustering of the first four data sets (Aggregation, Moons, Blobs-1000D and Spirals) result from an optimization using density based silhouette coefficient. Results for R15 and D31 were created using silhouette coefficient. The Flame data set was clustered using edge correlation.

measure has to be fixed as the optimization criterion. In our work we compared the use of silhouette coefficient and edge correlation as two such measures. However, both measures prefer convex shaped clusters and cannot adapt to all cluster shapes produced by DBSCAN. For this reason we proposed a density based interpretation of the silhouette coefficient, which rates the density of a cluster as the minimal eps-value it would be created with and sets it in relation to the minimal distance to the next cluster. In contrast to the original silhouette coefficient this optimization criterion can adapt to clusters of arbitrary shape. Fixing either $minPts$ or eps and rating the levels of the hierarchy results in the best value for the second parameter. The result can further be used as initialization for the next step of the alternating optimization process.

Our experiments showed that this method is capable of finding viable parameter combinations in a variety of cluster settings. The comparison of internal validation measures revealed that the density based silhouette coefficient performed best in most experiments. However in the search for convex shaped clusters, silhouette coefficient and edge correlation

performed better than the proposed density based silhouette coefficient.

Future work will focus on the analysis of the produced hierarchy. We expect other methods from hierarchical clustering to adapt well to our hierarchical DBSCAN variants. Iterating overt all possible eps values, i.e. every entry in the distance matrix can be cumbersome and slow down the optimization process. We expect that an initial binning of the distances will not deteriorate the results but speed up the process significantly. In our experiments, the density-based silhouette coefficient has shown to be very effective in finding non-convex but well separated clusters. A more detailed, theoretical analysis of this measure would be desirable.

REFERENCES

[1] R. Kruse, C. Borgelt, F. Klawonn, C. Moewes, M. Steinbrecher, and P. Held, *Computational Intelligence: A Methodological Introduction*, ser. Texts in Computer Science. New York: Springer, 2013.

[2] 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.

[3] 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.

[4] R. Winkler, "Prototype Based Clustering in High-Dimensional Feature Spaces," Ph.D. dissertation, Otto von Guericke University Magdeburg, 2015.

[5] J. Esmaelnejad, J. Habibi, and S. Yeganeh, "A Novel Method to Find Appropriate μ for DBSCAN," in *Intelligent Information and Database Systems*, ser. Lecture Notes in Computer Science, N. Nguyen, M. Le, and J. witek, Eds. Springer Berlin Heidelberg, 2010, vol. 5990, pp. 93–102.

[6] S. Vijayalaksmi, "A Fast Approach to Clustering Datasets using DBSCAN and Pruning Algorithms," *International Journal of Computer Applications*, vol. 60, no. 14, pp. 1–7, 2012.

[7] C. Braune, S. Besecke, and R. Kruse, "Density based clustering: Alternatives to DBSCAN," in *Partitional Clustering Algorithms*. Springer, 2015, pp. 193–213.

[8] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94)*, pp. 487–499, 1994.

[9] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[10] N. Bansal, A. Blum, and S. Chawla, "Correlation Clustering," *Machine Learning*, vol. 56, no. 1-3, pp. 89–113, 2004.

[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.