Variable Density Based Clustering

by Alexander Dockhorn, Christian Braune and Rudolf Kruse

Institute for Intelligent Cooperating Systems
Department for Computer Science, Otto von Guericke University Magdeburg
Universitaetsplatz 2, 39106 Magdeburg, Germany

Email: {alexander.dockhorn, christian.braune, rudolf.kruse}@ovgu.de

Contents

The DBSCAN clustering algorithm

- Density based clustering algorithm
- Parameters:
  - $\epsilon \rightarrow$ neighbourhood radius of each point
  - $min_{Pts} \rightarrow$ minimal number of neighbours for being core point

- Neighbourhood-set of a point consists of all points with distance less than or equal to ϵ

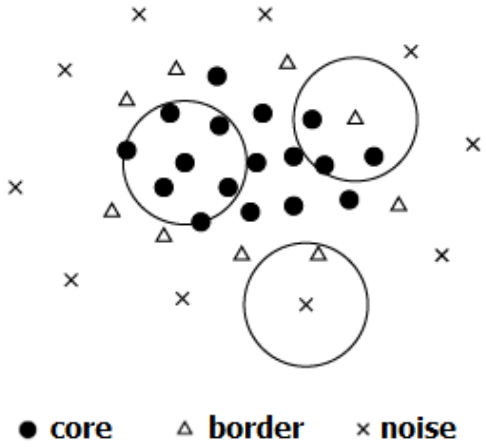
$$N_\epsilon(p) = \{ q \in D \mid d(p,q) \leq \epsilon \}$$

- Core-condition: If the size of a point's neighbourhood-set is greater than or equal to min_{Pts} the point is considered a core-point

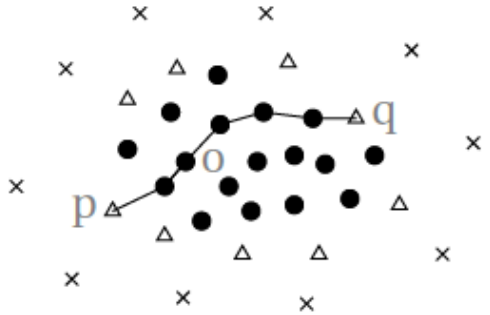
$$cores_{\epsilon, min_{Pts}} = \{ p \mid min_{Pts} \leq |N_\epsilon(p)| \}$$

Density-reachability and -connectedness

Cores, border points, noise



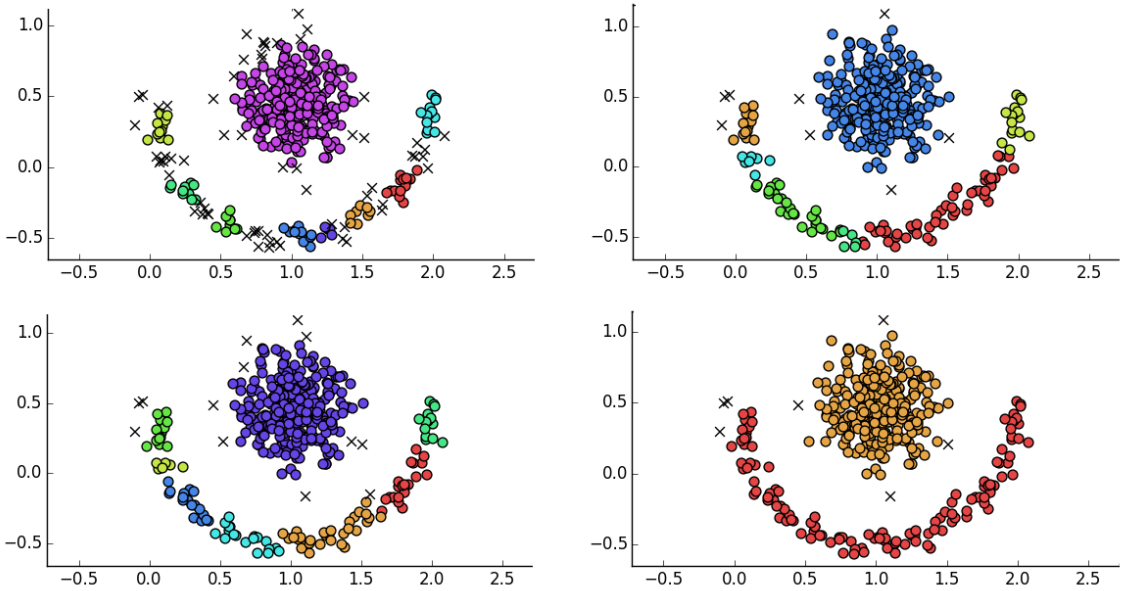
Density reachable and connected



- Border points are density-reachable by at least one core point
- Clusters are formed by the maximal set of density-connected points

One dataset, many clustering results

- Problem: Clustering algorithms depend on various parameters
- Clustering results of one algorithm using differing parameter initializations:



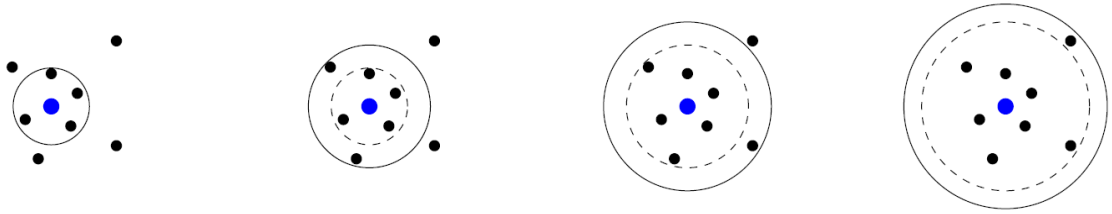
- Typically clustering validation techniques are used to rate the outcome and decide, which clustering will be used

What we have so far?

- We developed two variants of hierarchical DBSCAN (HDBSCAN) based on iterative parameter changes and their resulting cluster differences
- Monotonocity of parameter space can be used for efficient implementations of HDBSCAN
- Cluster Validation Indices can be used to find appropriate values of ϵ and min_{Pts}

Influence of ϵ on condition of a fixed min_{Pts}

- Increasing ϵ cannot decrease the neighbourhood-set size of a point.



- For two radii $\epsilon_1 \leq \epsilon_2$:

$$N_{\epsilon_1}(p) \subseteq N_{\epsilon_2}(p) \Rightarrow cores_{\epsilon_1,\, min_{Pts}} \subseteq cores_{\epsilon_2,\, min_{Pts}}$$

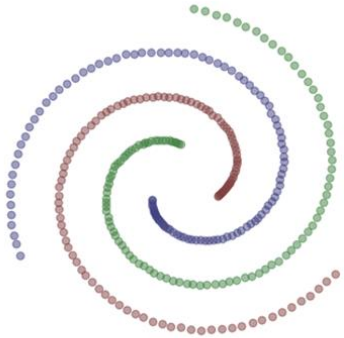
- Each entry $d(p,q)$ of the distance matrix represents an ϵ threshold for which a change of neighbourhood-sets occurs $\Rightarrow O(N^2)$ hierarchy level
- This does not need to change the clustering, since the pair (p,q) could already be density-connected

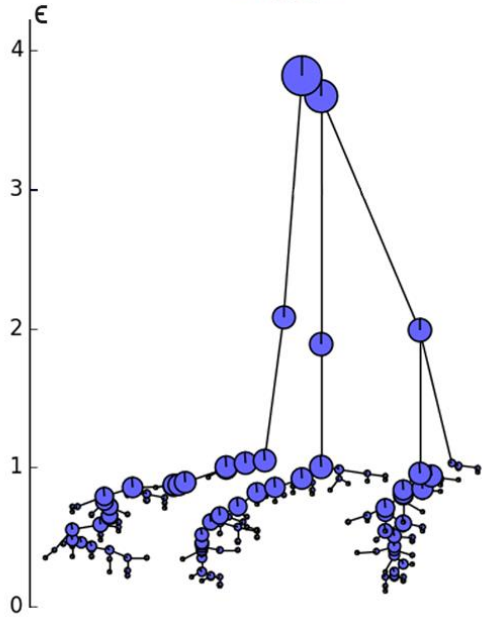
Hierarchical clustering iterating ϵ

- Iterate through all entries of the distance matrix
- Sort matrix in ascending order to build hierarchy bottom-up

Algorithm 1: min_{Pts}-HDBSCAN

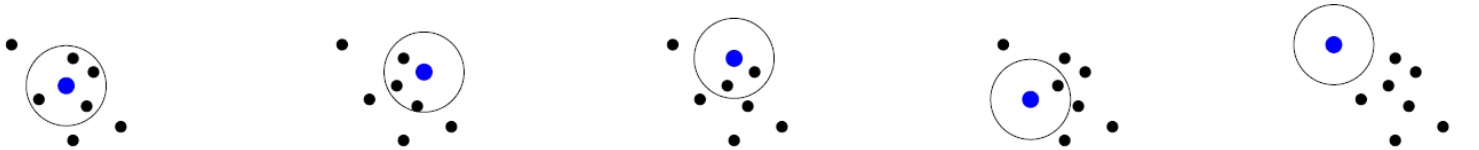
1 Fix parameter min_{Pts}
2 Sort distance matrix ascendingly (x, y, r)
3 **For** $(x, y, r) \in$ sorted distance matrix **do**:
4 update neighbourhood-set of x and y
5 update clustering
6 **if** clustering changed **then:**
7 add clustering to hierarchy
8 **End For**





Influence of min_{Pts} on condition of a fixed ϵ

- Decreasing min_{Pts} cannot decrease the number of cores



$min_{Pts} = 5$ $min_{Pts} = 4$ $min_{Pts} = 3$ $min_{Pts} = 2$ $min_{Pts} = 1$

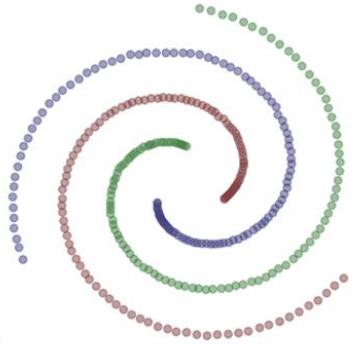
- For two thresholds $min_{Pts1} > min_{Pts2}$

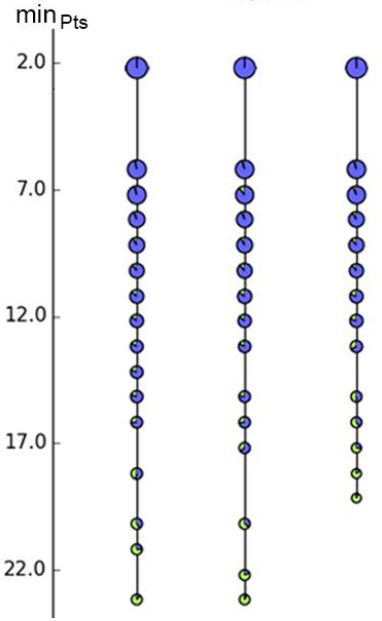
$$cores_{\epsilon, min_{Pts1}} \subseteq cores_{\epsilon, min_{Pts2}}$$

- Since the neighbourhood-set of a point can at most consist of every point in the dataset, the maximum number of hierarchy levels is N

Hierarchical clustering iterating min_{Pts}

- Iterate through all neighbourhood-set sizes





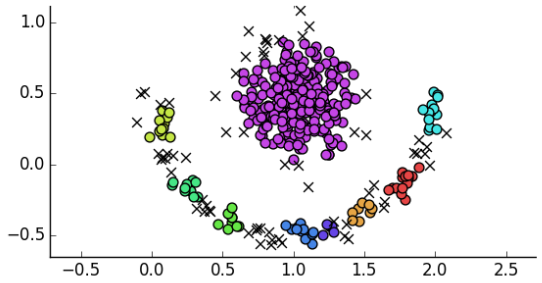
Algorithm 2: ϵ-HDBSCAN

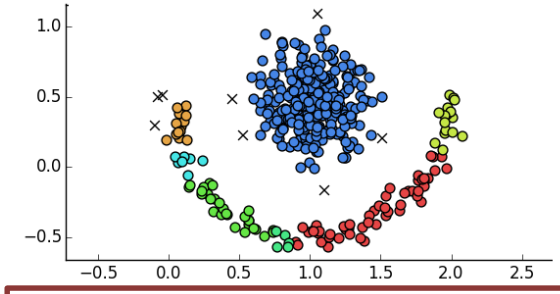
```
1 Fix parameter ε
2 Calculate neighbourhood-sets
3 For min_Pts from N to 1 do:
4     update density-connectedness
5     update clustering
6     if clustering changed then:
7         add clustering to hierarchy
8 End For
```

From last years method

- Problem: Clustering algorithms depend on various parameters

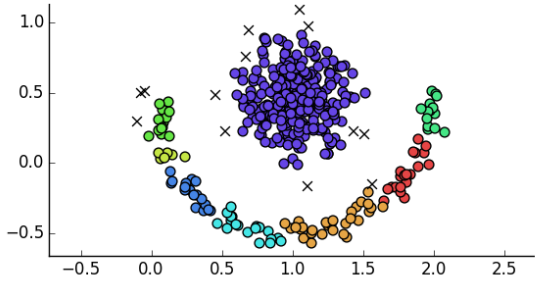
$\epsilon = 0.1$
$min_{Pts} = 8$

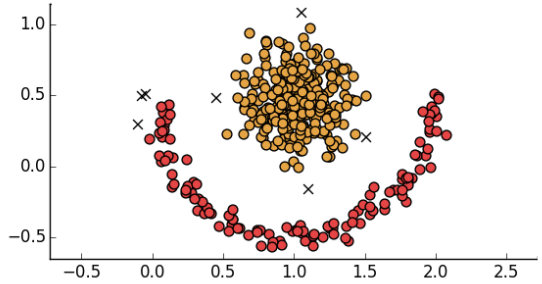




$\epsilon = 0.13$
$min_{Pts} = 8$

$\epsilon = 0.1$
$min_{Pts} = 5$



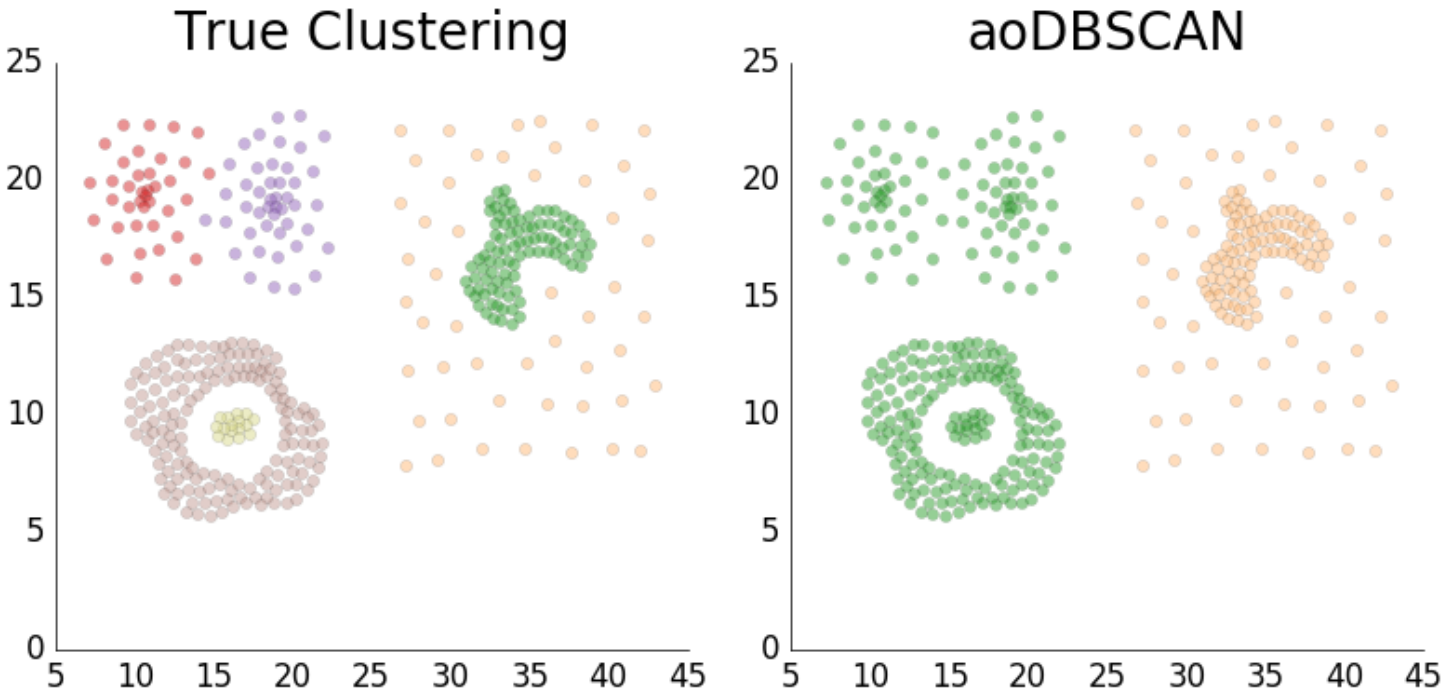


$\epsilon = 0.13$
$min_{Pts} = 5$

- AO-DBSCAN partially solves the problem of estimating appropriate parameters

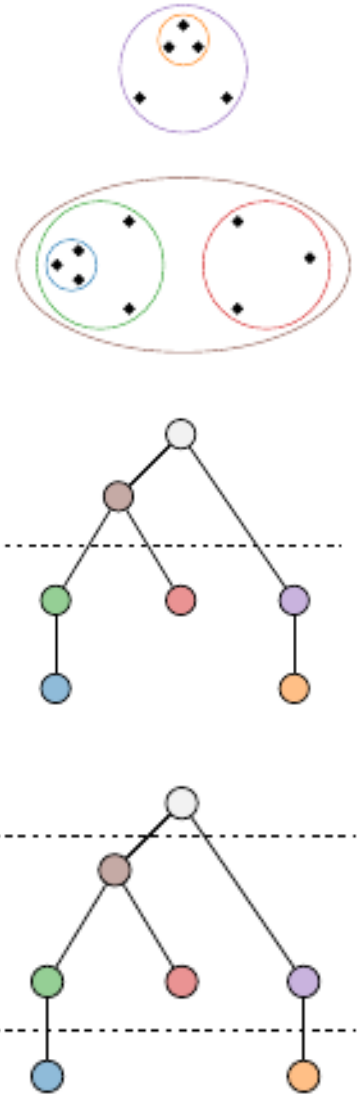
The problem of differing density clusters

- However, it fails in the presence of differing density clusters!

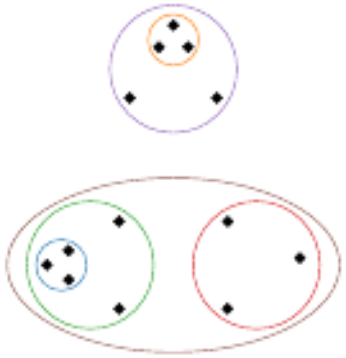


Why does this happen?

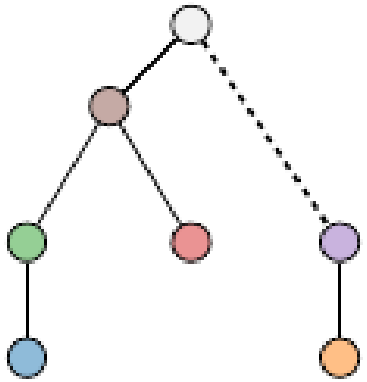
- AODBSCAN is limited to horizontal cuts of the hierarchy
- Those resemble a constant combination of ϵ and min_{Pts} for all clusters
- However, sometimes a hierarchy of clusters is more appropriate for the data set
- Although, the full hierarchy contains to many levels
- Problem: How to filter the hierarchy for variable density clusters?



A) Parameter Changes



- The hierarchies created by HDBSCAN contain information about the parameter space
- Huge gaps between consecutive levels indicate large parameter changes
- This can be compared with an cost-based approach
  - Cost = how much do I have to adjust a parameter for the next merge
- Smooth density transitions will not trigger
  - See example to the right



A) Parameter Change Cut

- For each edge:
  - Compute hight difference = parameter difference
- For the edges with the highest difference:
  - Add bottom level node to the filtered hierarchy
- A point always belongs to the node with the highest density it is assigned to

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*

B) Estimating the density of a cluster

- Density is defined by the number of mass per unit volume
- This corresponds to the number of points per area size of the cluster
- Problem: How do we get an appropriate estimate of the clusters area / volume? How can we neglect empty space from this estimate?
- Solution: Using shape descriptors for estimating the area.
  - In this work we used Alpha Shapes

B) Alpha Shapes

- Alpha shapes produce non-convex hulls for an arbitrary set of points
- For alpha = ∞ the alpha shape resembles a convex hull
- The alpha shape degenerates for small alphas

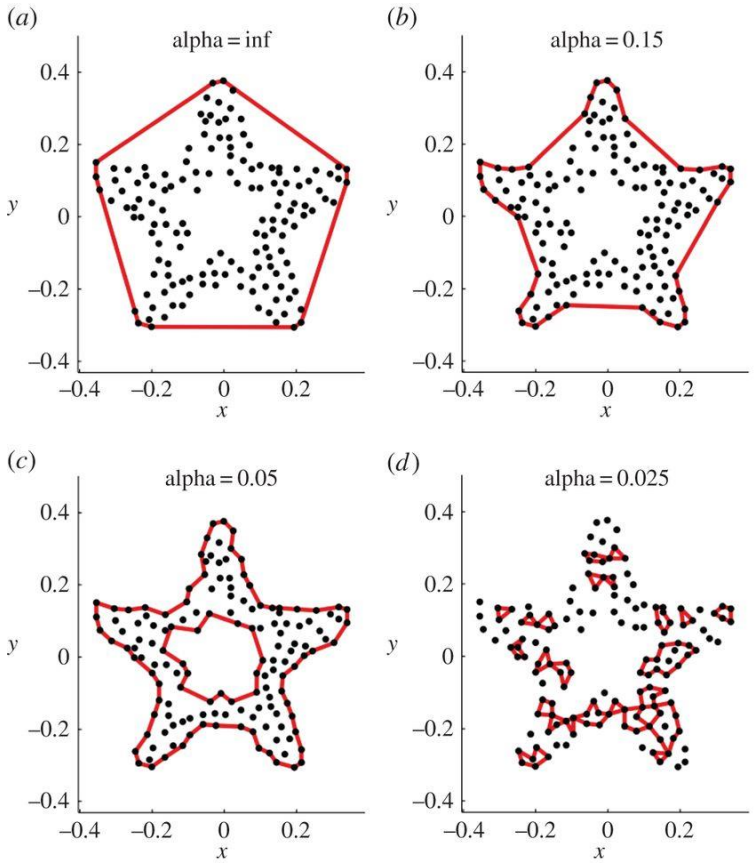


Image from:
Brassey, C. A., & Gardiner, J. D. (2015). An advanced shape-fitting algorithm applied to quadrupedal mammals: improving volumetric mass estimates.
Royal Society Open Science, *2*(8), 150302.

B) Alpha Shape Cut

- For each edge:
  - Compute the area before and after the merge
- For the edges with the highest area difference:
  - Add bottom level node to the filtered hierarchy
- A point always belongs to the node with the highest density it is assigned to

Algorithm 2 Alpha Shape Cut

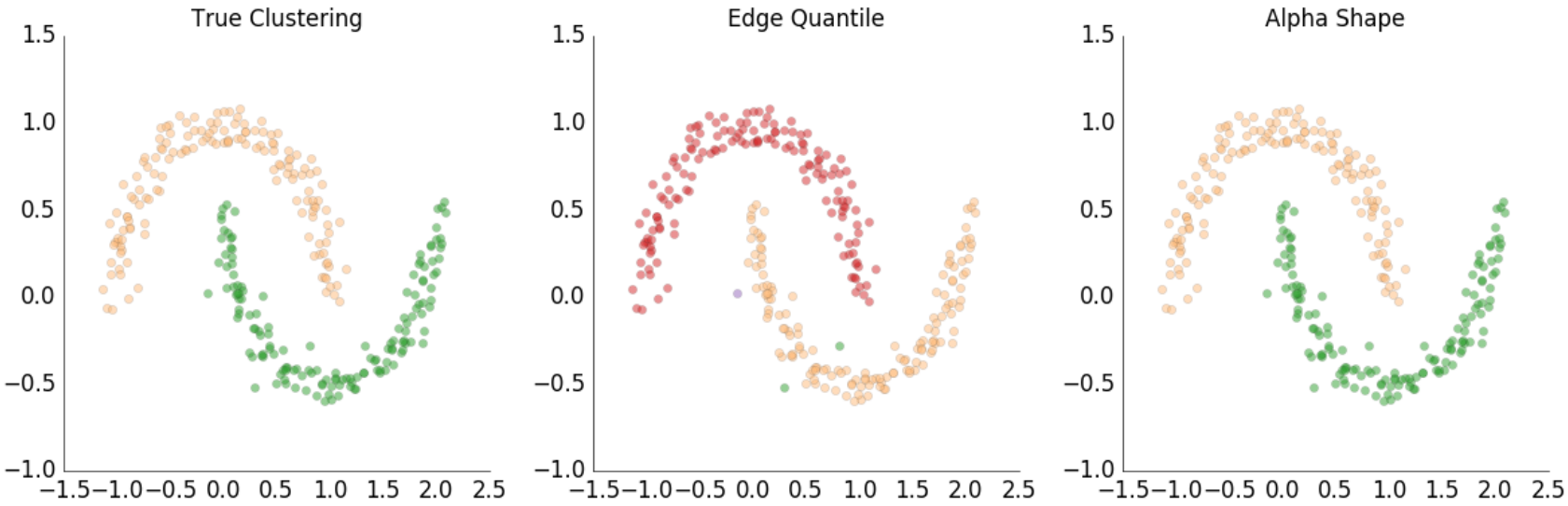
Input: G = DBSCAN hierarchy, quantile

```
cutlist ← list()
for all (u, v) ∈ 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 ∈ Points do
    set labels[p] to the index of the first
    cutlist element it is part of
end for
return labels
```

Moons Data Set

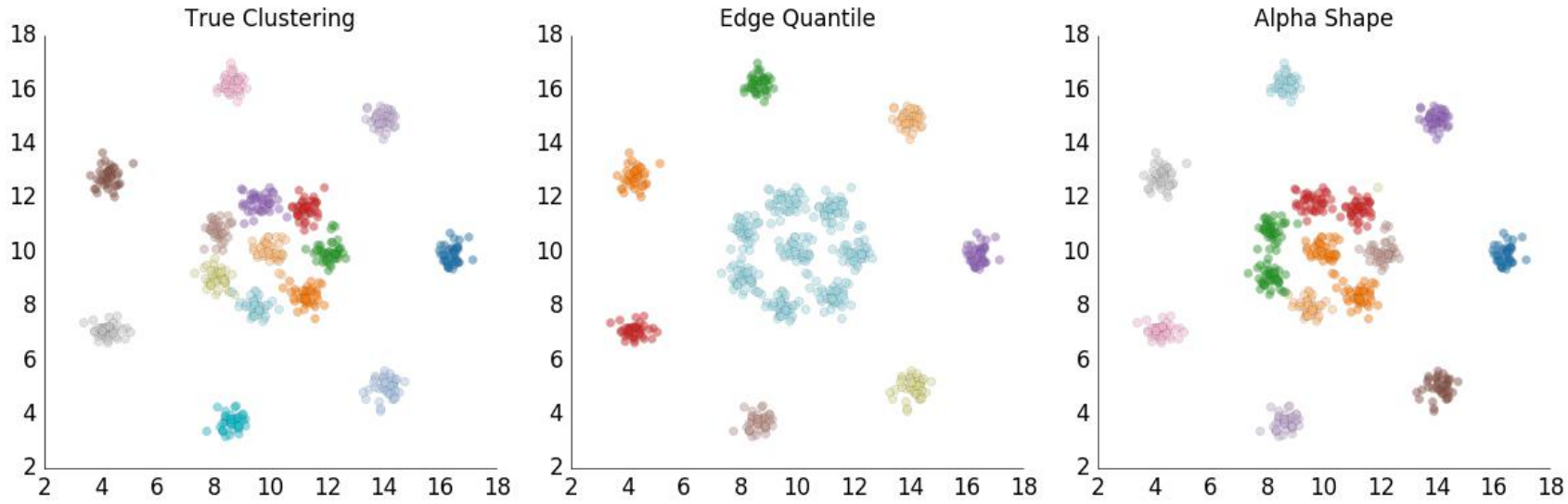
- Typical example for density based clustering



- Parameter Change Cut is sensitive to single points
- Alpha shape is more robust, since the clusters area is not influenced by single noise points

R15 Data Set

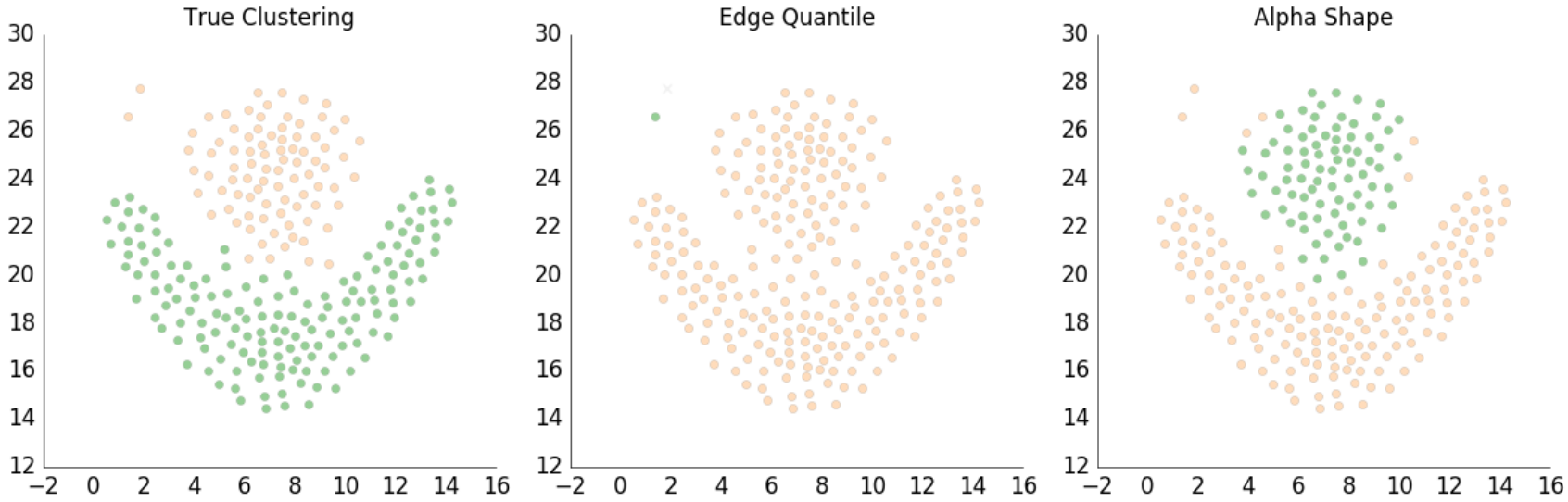
- Varying degrees of cluster separation



- A fixed quantile cannot always detect all relevant merges. A more sophisticated distribution analysis might overcome this problem.
- Alpha Shape Cut performed better in detecting merges of multiple clusters.

Flame Data Set

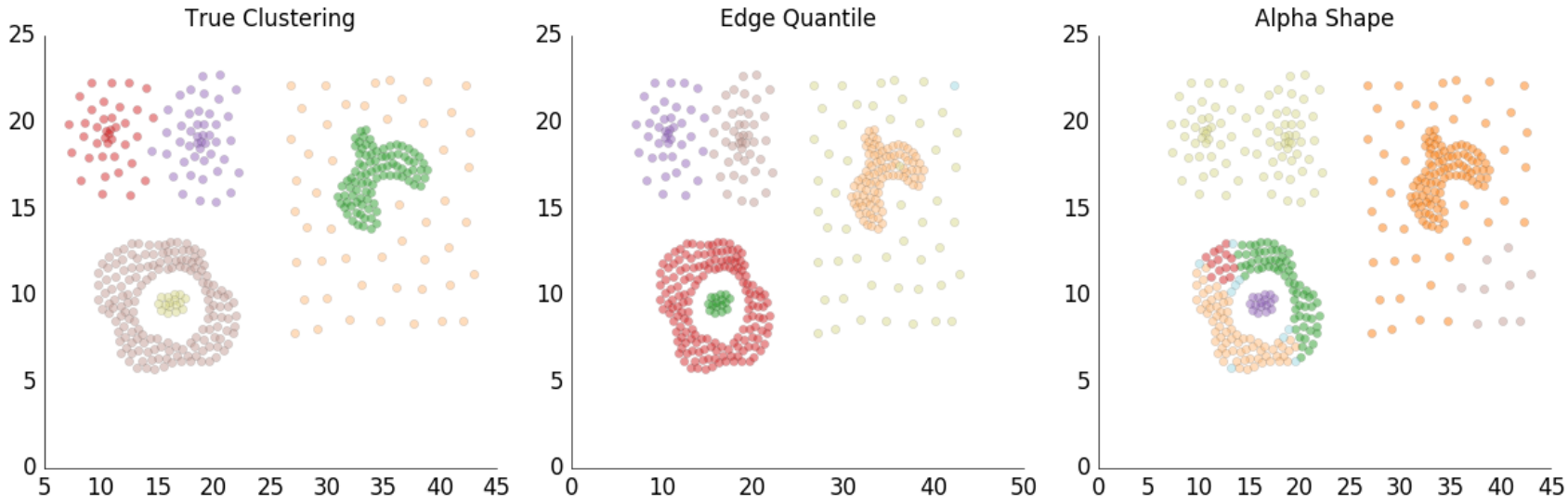
- Smooth density transitions



- The Edge distribution gets skewed by outliers on the top left. Parameter change cut therefor fails in determining an appropriate cut value.
- Alpha Shape Cut recognizes the large merge of the two central clusters.

Compound Data Set

- Nested cluster structures and clusters of varying density and shape



- Parameter Change Cut is able to find separations of fluent cluster merges
- Alpha Shape Cut fails in this scenario

Conclusion

- Clusters of variable density can be extracted from HDBSCAN hierarchies
- While both non-horizontal cuts do not always perform well, it is a great help for interactive data analysis methods.
- The single parameter (cut-value) is monotone in its behaviour and therefore easy to adjust

- Parameter Changes between a merge of clusters can be to small
- Area estimate is much more robust for cluster merges, but fails in other scenarios
  - No free lunch!

Suggestions for future work

Current Problems	Possible solutions
Parameter changes converge to zero in high dimensional datasets	MST streaming algorithm for HDBSCAN
Cuts based on Alpha shapes and CLASH are currently only implemented for 2D datasets	Extend Area calculation to hyper-volume calculation
Combine capabilities of AO-DBSCAN and non-hierarchical cuts	Local parameter estimates
Single Outliers can skew the distribution of parameter changes	More sophisticated distribution analysis for reducing this influence

Thank you for your attention!

Download it at:
http://fuzzy.cs.ovgu.de/wiki/pmwiki.php/Mitarbeiter/Dockhorn

by Alexander Dockhorn, Christian Braune and Rudolf Kruse

Institute for Intelligent Cooperating Systems
Department for Computer Science, Otto von Guericke University Magdeburg
Universitaetsplatz 2, 39106 Magdeburg, Germany

Email: {alexander.dockhorn, christian.braune, rudolf.kruse}@ovgu.de