Data Science and Engineering (2020) 5:360–374
https://doi.org/10.1007/s41019-020-00137-x

“Show Me the Crowds!” Revealing Cluster Structures Through AMTICS

Florian Richter[1] · Yifeng Lu[1] · Daniyal Kazempour[1] · Thomas Seidl[1]

Received: 12 May 2020 / Revised: 10 July 2020 / Accepted: 16 July 2020 / Published online: 11 August 2020

Abstract
OPTICS is a popular tool to analyze the clustering structure of a dataset visually. The created two-dimensional plots indicate very dense areas and cluster candidates in the data as troughs. Each horizontal slice represents an outcome of a density-based clustering specified by the height as the density threshold for clusters. However, in very dynamic and rapidly changing applications, a complex and finely detailed visualization slows down the knowledge discovery. Instead, a framework that provides fast but coarse insights is required to point out structures in the data quickly. The user can then control the direction he wants to put emphasize on for refinement. We develop AMTICS as a novel and efficient divide-and-conquer approach to pre-cluster data in distributed instances and align the results in a hierarchy afterward. An interactive online phase ensures a low complexity while giving the user full control over the partial cluster instances. The offline phase reveals the current data clustering structure with low complexity and at any time.

Keywords Data streams · Hierarchical clustering · Density-based · Visual analysis

1 Introduction

Clustering is an essential task in the field of data mining and unsupervised machine learning. The initial data explorations are usually an important but difficult and exhausting step of the analysis. A huge performance bottleneck is the identification of parameter ranges to return interesting results.

In density-based clustering approaches, we are mainly interested in finding dense regions of neighboring objects. However, the definition of proximity has to be chosen mostly manually while investigating a new dataset. Probably, the most prominent density-based clustering method is DBSCAN [4], which detects arbitrarily shaped clusters with similar densities while being robust against noise. Estimating proper parameters to distinguish between sparse and dense regions is a nontrivial task here, especially in dynamic applications.

A method called OPTICS [2] was designed to cope with this problem. As an extension of DBSCAN, it provides a hierarchical model of the cluster structure in the dataset. The results are represented as a two-dimensional plot, such that analysts visually identify troughs as cluster candidates.

While being a suitable visualization tool for experts to estimate the number and size of clusters hidden in the dataset, it still contains the risk of poor parameter choices. In the worst-case scenario, no troughs can be identified due to a wide spectrum of densities in the dataset, leading to a flat OPTICS plot. Our novel approach AMTICS provides an interactive way to choose promising density levels for further investigation. Due to this explicit choice, attention on certain aspects is established and the coarser presentation of the dataset offers a broader comprehension of the data. The advantage is not only the efficient re-computations to allow a stream application of AMTICS. Instead of visualizing all minor density fluctuations, our novel method AMTICS provides a coarse estimation of the cluster structure for a fast explorative human-based visual analysis. An ensemble of density-based online clustering instances is the core of AMTICS. These efficient and distributed instances are interchangeable, so observation levels can be added or discarded. Analysts are free to choose which density levels are promising to increase the granularity there. This interactivity improves the analysis performance of the human-in-the-loop. For changing

✉ Florian Richter
richter@dbs.ifi.lmu.de

Yifeng Lu
lu@dbs.ifi.lmu.de

Daniyal Kazempour
kazempour@dbs.ifi.lmu.de

Thomas Seidl
seidl@dbs.ifi.lmu.de

[1] LMU Munich, Munich, Germany

conditions or new analysis interests, the observation focus can be changed dynamically. In the final step, all instances are aligned to produce an approximated density plot of the recently observed objects which is a huge benefit for any further data analysis. AMTICS, as shown here, utilizes DenStream. However, the main contribution is the agglomeration of approximative online density-based clustering results, so DenStream can be replaced by other clustering techniques.

The research focus is the revelation of cluster structures in dynamic environments. The result is by far not a certain clustering, but serves as a hint to start a more thoroughly cluster analysis or concept drift detection. Density-based clustering is a wide field of different paradigms, which yield various variants of clustering instances. The identification of suitable parameters for succeeding cluster techniques is highly depending on application and the focus of the analysis, and not covered in this work.

2 Preliminaries

Density-based clustering is a well-studied topic in data science. As most readers will already know, density is here defined by two parameters: the radius ε to define the neighborhood of each point $N_\varepsilon(x)$ and the minimal number of points MinPts required for a dense neighborhood. Every point is

- A core point if it has a dense neighborhood. Neighbored core points establish clusters.
- A border point if its neighborhood contains a core point. It is also added to this core point's cluster.
- Noise otherwise.

One of the most popular density-based methods is DBS CAN [4]. It selects points until a core point is found. All transitively neighboring core points are merged to a common cluster, and neighboring border points are included. If no further reachable core points can be found, this strategy is repeated for the remaining yet untouched points until all points are classified as either core points, border points or noise. A major benefit of DBSCAN is its ability to detect arbitrarily shaped clusters, while many other clustering approaches focus on elliptically shaped clusters. Second, it also includes robustness against noise due to the density property. The fixed ε-parameter, on the other hand, is a drawback as clusters with deviating densities are not detected in a single DBSCAN instance. Choosing a lower ε value will assign sparse clusters to noise, while a higher ε value will more likely merge separate nearby clusters.

To overcome this issue and to assist in finding a suitable ε value in case of an initial data exploration task, Ankerst et al. developed OPTICS [2]. Given MinPts, this method determines for each point its core distance, the minimal distance needed such that the ε-neighborhood contains MinPts many points. For a point p, let kNN be the kth nearest neighbor and d a distance function. Then, the core distance is defined by

$$\mathrm{core}_{\varepsilon,\mathrm{MinPts}}(p) = \begin{cases} d(p, \mathrm{MinPtsNN}), & |N_\varepsilon(p)| \geq \mathrm{MinPts} \\ \text{undefined}, & \text{otherwise} \end{cases}$$

The ε value is used as an upper bound for performance improvement. Using the core distance, the reachability distance can be defined as

$$\mathrm{reach}_{\varepsilon,\mathrm{MinPts}}(o, p) = \begin{cases} \max(\mathrm{core}_{\varepsilon,\mathrm{MinPts}}(p), d(p, o)), & \text{if } |N_\varepsilon(p)| \geq \mathrm{MinPts} \\ \text{undefined}, & \text{otherwise} \end{cases}$$

The set of data points gets ordered by its reachability distance. For each point, its successor is the point with the smallest reachability distance out of the unprocessed points. This ordering is not unique, due to start point ambiguity and potential choices between equidistant objects. Finally, a reachability plot is provided using the ordering on the x-axis and the reachability distance on the y-axis. Since dense object clusters in the data space have low pairwise reachability distances, they are accumulated in the plot and the cluster is identified as a trough in the reachability plot.

In an interactive online setting, we cannot apply OPTICS due to its high computational complexity. Hence, we propose a novel approach, which adapts the idea of DenStream [3] by utilizing micro-clusters. A micro-cluster is an aggregation of a group of data points, storing the number of aggregated points as the weight w, the center of the group c and the radius r. To enable incremental updates, instead of storing the center and radius, a linear sum LS and a squared sum SS are stored. For an update of a micro-cluster with point p, the procedure

$$w = w + 1, \mathrm{LS} = \mathrm{LS} + p, \mathrm{SS} = \mathrm{SS} + p^2$$

has to be performed. As a decay mechanism, the current values are determined after multiplying all three parameters with the factor $2^{-\lambda*\delta t}$ if δt is the time interval since the last update of the micro-cluster. $\lambda > 0$ has to be chosen to suit the desired rate of decay. The center and the radius can be derived from the provided statistics as $c = \mathrm{LS}/w$, $r = \sqrt{(\mathrm{SS}/w^2 - \mathrm{LS}^2/w)}$.

The radius is defined as the standard deviation of the aggregated points. DenStream uses two sets of micro-clusters: the outlier micro-clusters and potential micro-clusters. Outlier micro-clusters contain few points such that their weight is below a certain threshold. If it decays without new points being merged into this cluster, it will disappear. Exceeding the weight threshold, it will become a potential micro-cluster. For the final cluster result, only the p-micro-clusters are used by merging touching p-micro-clusters into macro-clusters.

3 AMTICS

Our clustering approach is a two-phase algorithm maintaining an online intermediate representation of clusters and providing an offline refinement step to construct the current cluster hierarchy. The online phase uses multiple instances of a density-based online clustering algorithm for various density levels $\varepsilon_1, \ldots, \varepsilon_k$. Traditionally, OPTICS is applied first and interesting density levels are determined visually. In this work, we reverse the application order by the application of cluster algorithms on different density levels and merging this information into an approximate OPTICS plot. The key points are the ensemble of single-density clusterings, the alignment of micro-clusters and the final transformation into a reachability plot.

3.1 Online Stream Ensemble

We choose DenStream [3] as a starting point. Few required parameters make it suitable for user interaction. The maintained finite set of micro-clusters is necessary for the complexity constraints. Further, it allows to compare micro-cluster structures of different density levels, such that clusters can be aligned in one model. In future work, we will investigate which density-based online cluster methods, for example a grid-based approach, can be used instead of DenStream but this is not the focus in this work.

At all time, a finite set of DenStream instances $\{\mathrm{DS}_\varepsilon \mid 0 \le \epsilon \le \infty\}$ observe the stream and maintain their micro-clusters. Each instance can be deleted or initialized anytime during the stream except of two instances: DS_0 and DS_∞. DS_0 will classify every object as a different one-point cluster. DS_∞ builds exactly one large cluster containing all objects. Both instances define the boundaries of our final result.

Since DenStream guarantees to maintain a finite set of micro-clusters, keeping several but finitely many instances is within the complexity limitations of an online algorithm. In case that the user wants to introduce a new instance, we duplicate the denser neighboring instance. Due to the decay λ, the new instance will quickly adapt to the recent points. Each ε-instance is a set of overlapping micro-clusters. Two micro-clusters are touching or overlapping if the distance between their centers is smaller than the sum of their radii, which is the standard deviation of its points. We show the intermediate result of four instances in Fig. 1 for the two moons dataset. From left to right and from top to bottom, the ε level is decreasing. Note that the first instance contains always only one large cluster and the last one contains no cluster. Although the plot contains both types of micro-clusters, only the red potential micro-clusters are used for the following steps.

Algorithm 1: AMTICS.getClusters

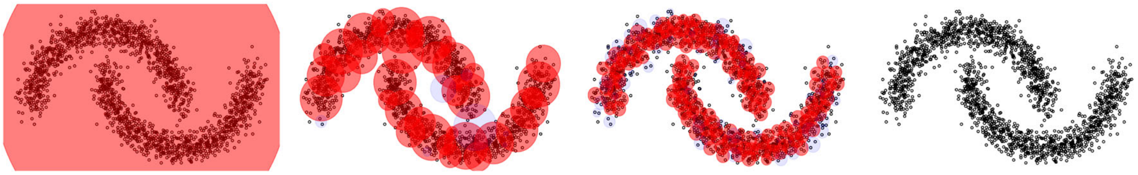
```
Data: Set of DenStream instances $DS_\varepsilon$
Result: Mapping on micro-cluster sets $m : (\varepsilon, i) \rightarrow MC$
initialization of empty mapping $m$;
foreach $\varepsilon$ do
    $i = 0$;
    $C = \emptyset$;
    while $DS_\varepsilon$ contains potential micro-clusters do
        $i = i + 1$;
        get any potential micro-cluster $mc \in DS_\varepsilon$;
        $C$ = find all p-micro-clusters connected with $mc$;
        remove all micro-clusters in $C$ from $DS_\varepsilon$;
        $m(\varepsilon, i) = C$;
    end
end
```

3.2 Hierarchical Alignment

The previous online phase provides layers of clusterings for all ε-values, and we need to align the clusters of each layer with the clusters of the layer below. A cluster is represented by a set of micro-clusters, which are points with a certain weight. We call two clusters C_1 and C_2 directly related if $C_1 \in \mathrm{DS}_{\varepsilon_1}$ and $C_2 \in \mathrm{DS}_{\varepsilon_2}$ with $\varepsilon_1 > \varepsilon_2$, there is no instance in between $\mathrm{DS}_{\varepsilon_1}$ and $\mathrm{DS}_{\varepsilon_2}$, and $D(C_1, C_2) = \min(\{D(C_1, C_i) \mid C_i \in \mathrm{DS}_{\varepsilon_2}\})$ for a suitable distance function D.

We model the alignment of two instances in the refinement as a transportation model, since we have to match micro-clusters of different weights and positions. Therefore, we apply the Earth Mover‘s Distance EMD [11] and extend it to compute the cluster distance.

A function for this set-to-set distance has to be chosen carefully as it highly determines the result. Popular candidates for set distances are single-link, complete-link and average-link distances which are used in agglomerative hierarchical clustering. Since our approach also constructs a cluster hierarchy, we could apply AHC here as well. The major issue is that we do not deal with exact object sets but with approximative partitions. This fuzziness of object membership between different dendrogram layers causes these distance functions to be not suitable in this application. An applicable distance should combine two properties: spatial distance and cluster weight difference. Also the weight of a larger cluster could be divided onto several smaller clusters. This is more similar to the task of optimal transport.



(a) $\varepsilon = \infty$ **(b)** $\varepsilon = 0.3$ **(c)** $\varepsilon = 0.1$ **(d)** $\varepsilon = 0.0$

Fig. 1 Two moons dataset clustered with different instances of DenStream. Potential micro-clusters are drawn in red, outlier micro-clusters in blue

The Earth Mover's Distance EMD [11] is well suited for this task as it calculates the cost of shifting weights from one set of bins to another set of empty bins. The bins represent the micro-cluster centers, and the distance between two bins is the distance between both centers. The EMD covers the case that a large cluster is split into several smaller clusters in the lower ε-level, which is rather common in our application.

The EMD is defined for two clusters $C_1 = \{(p_1, w_1), ..., (p_m, w_m)\}$ and $C_2 = \{(q_1, v_1), ..., (q_n, v_n)\}$. We use the Euclidean distance $d(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$ as ground distance between two micro-cluster centers. The aim is to find the flow $F = (f_{i,j}) \in \mathbb{R}^{m \times n}$ of weight from C_1 to C_2 which minimizes the costs given by

$$c_F = \sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j} d(p_i, q_j)$$

in strict accordance with the following constraints:

- The flow has to be nonnegative, so weights are only sent from C_1 to C_2 and not vice versa: $f_{i,j} \geq 0,\ \forall 1 \leq i \leq m, 1 \leq j \leq n$
- The sent flow is bounded by the weights in C_1: $\sum_{j=1}^{n} f_{i,j} \leq w_i,\ \forall 1 \leq i \leq m$
- The received flow is bounded by the weights in C_2: $\sum_{i=1}^{m} f_{i,j} \leq v_j,\ \forall 1 \leq j \leq n$
- All weights possible have to be sent: $\sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j} = \min\left(\sum_{i=1}^{m} w_i, \sum_{j=1}^{n} v_j\right)$

The distance is then defined as $\mathrm{EMD}(C_1, C_2) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j} d(p_i, q_j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j}}$.

We apply the EMD on all potential candidate pairs which are clusters of consecutive density levels. Eventually, we aim for a tree of clusters such that the root is the cluster in DS_∞ containing all points. The height of each node represents the density level of this cluster. If two clusters are directly related, then the corresponding tree nodes are connected. Our method determines for every cluster the parent cluster with the minimal EMD. In Fig. 2, a matching is performed for two ε layers. The strong lines indicate the best matching parent cluster. After application of the EMD, we gain an alignment of all pairwise consecutive cluster layers connected by the smallest EMD distances. The desired outcome is a tree-like hierarchy, such that child nodes contain always less points than parent nodes, which is comparable to a max-heap. The reason is discussed in the following.

Algorithm 2: AMTICS.buildHierarchy

```
Data: Mapping on micro-cluster sets $m : (\varepsilon, i) \rightarrow MC$
Result: Hierarchy of clusters H
initialize empty graph H
sort all $\varepsilon$ descending
/* create node N for $m(\infty, 0)$ */
N = node($m(\infty, 0)$)
parentNodes = list(N)
H.root = N
/* Generate a graph of related cluster nodes */
foreach $(\varepsilon_{prev}, \varepsilon_{next})$ do
    childNodes = list()
    foreach $m(\varepsilon_{next}, i)$ do
        $node_c$ = node($m(\varepsilon_{next}, i)$)
        foreach $node_p \in parentNodes$ do
            $N.d(node_p) = EMD(node_c, node_p)$
        end
        /* find closest parent nodes */
        $node_{p1} = min(N.d)$
        $node_{p2} = min_{second}(N.d)$
        if $N.d(node_{p1}) \approx N.d(node_{p2})$ then
            $merge(node_{p1}, node_{p2})$
        end
        Add $(N, node_{p1})$ to $H$
    end
    parentNodes = childNodes
end
```

3.3 Shared Micro-cluster Coverage

In an optimal case, large clusters split into smaller shards on the lower ε-levels and the constructed hierarchy represents a relation of subsets regarding the contained point set and the cluster nodes represent a max-heap structure regarding the cluster weights. For a maximal ε value, all points are contained in a super-cluster which is also true for DS_∞. The lowest level classifies each point as noise and so does

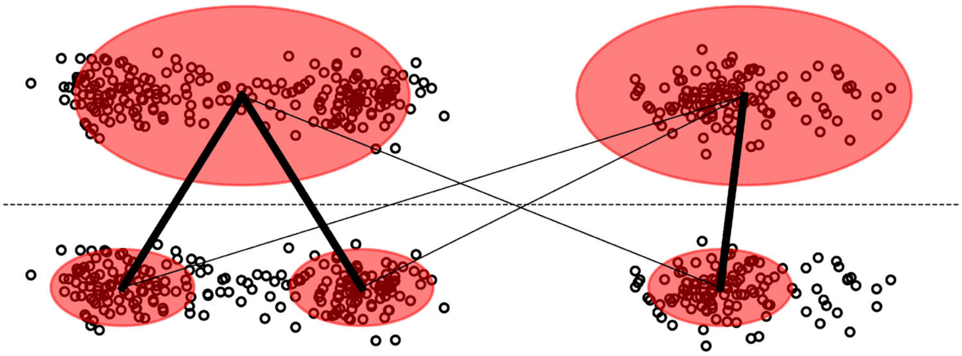


Fig. 2 Distances for all pairs of clusters of consecutive ε levels have to be computed to find the closest and most likely parent cluster

DS_0. This is not necessarily the case at this point, even for small synthetic datasets. The generation of micro-clusters is depending on the processing order of the points, and we usually cannot guarantee a perfectly uniform distribution of stream objects, neither spatial nor temporal.

While OPTICS and DBSCAN consider ε neighborhoods for all points, DenStream trades this accuracy for its ability to aggregate points into micro-clusters. The more complex the dataset is and the higher the dimension of the points is, the more likely is that the aligned micro-clusters do not form a hierarchy anymore. We identified two effects causing problems here by violating the heap condition, which is the case if a cluster in a smaller ε level is the children of a cluster with less weight. To ensure a valid hierarchy, clusters with larger neighborhoods have to cover clusters with smaller ε values.

As we do not store every singular point, we assume the contained points to be mostly equally distributed within the defined circular area. Although this is a very useful assumption for the general case, it has a drawback. Every point added to the micro-cluster shifts the center toward this point's coordinates. If all points are perfectly equally distributed regarding not only their position but also their sequential occurrence, the micro-cluster would not move. In reality and even for synthetic datasets, a small set of points can cause a cluster to shift apart from its potential connecting neighbor, causing a cluster to split. We call such events micro-trends, and they are the more likely, the larger a dataset is.

The actual problem occurs by comparing micro-clusters of different radii which we do by aligning micro-clusters of different ε levels. Larger micro-clusters are much more affected by micro-trends. Let us assume two one-dimensional micro-clusters $mc_1 = \{0, \ldots, 7\}$ and $mc_2 = \{8, \ldots, 15\}$ as displayed in Fig. 3. The micro-clusters are touching so they form a cluster in the output result. If we add three additional points per micro-cluster, the micro-clusters lose their connection as the points are not equally distributed anymore. In comparison with these large micro-clusters, we clustered the same dataset with smaller micro-clusters. Each of the four micro-clusters contains four neighboring points. When adding the six additional points, the outer micro-clusters absorb three points each. However, the points within all clusters are equally distributed, so the centers are not shifted. The radius of each outer micro-cluster is reduced but all four micro-clusters are still connected.

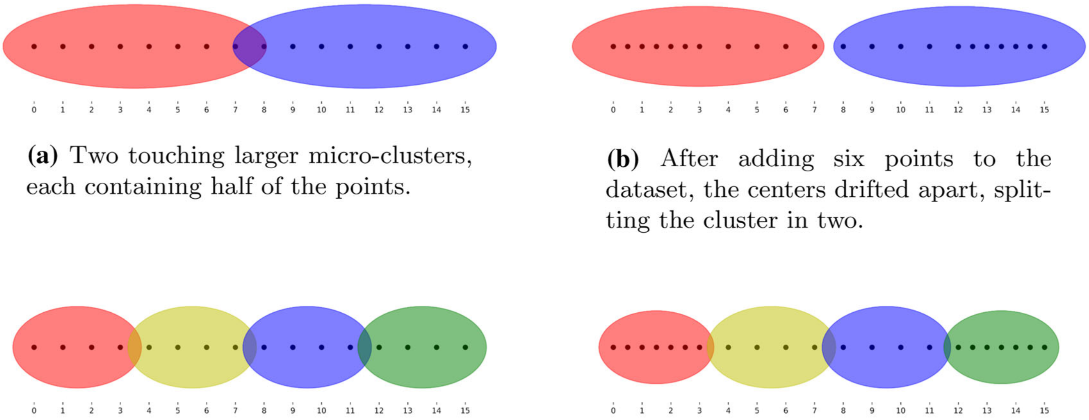
In Fig. 4, we display two DenStream instances where the previously described effect occurs. Considering both gaps with smallest distance between the moons, the complete point set is connected in the lower instance $DS_{0.25}$ while disconnected in $DS_{0.3}$.

Starting from the initial assumption that smaller clusters with higher density on the lower hierarchy levels should always stay connected in higher ε levels, we suggest the following strategy. During the creation phase of the cluster nodes, we previously determined for each cluster the closest parent cluster considering the EMD. Instead, we also compute the distance to the second nearest potential parent cluster. If the ratio of the nearest p_1 to the second nearest cluster p_2 as $ratio = \mathrm{EMD}(p_1, n)/\mathrm{EMD}(p_2, n)$ is close to 1.0, both parent clusters cover a dominant area of the child cluster.

We merge this pair of clusters by shifting all the weight and micro-clusters of the second one p_2 to the first one p_1. In addition, all pointers from and to p_2 have to be changed accordingly such that the parent node of p_2 becomes p_1. Possibly, distances have to be recalculated as p_1 contains more micro-clusters now. It is also possible that this initializes a cascade of merging operations bottom-up until the root node is reached. This is repeated top-down until no merges are required anymore. This cascade is still no issue for the online applicability as the number of operations is limited by the number of ε layers above, which is a user-defined finite and mostly small number. The algorithmic description is given in Algorithm 3.

3.4 Local Outliers

Border points sometimes are not covered by potential micro-clusters of larger radius while being contained in smaller micro-clusters, see, e.g., Fig. 5. Two DenStream instances are applied to this dataset with ε values of 1.1 and 0.8. As



(a) Two touching larger micro-clusters, each containing half of the points.

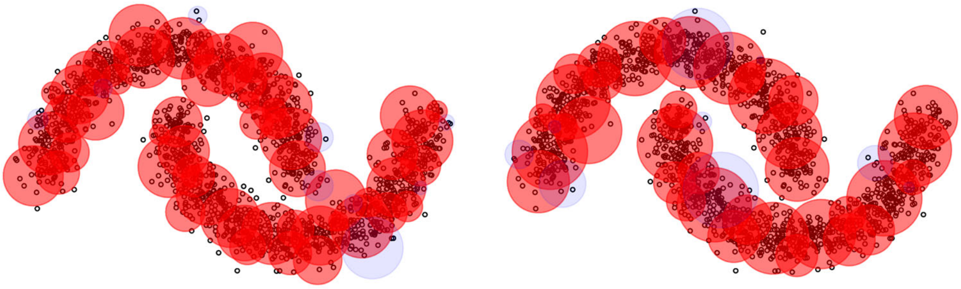
(b) After adding six points to the dataset, the centers drifted apart, splitting the cluster in two.

(c) Four touching smaller micro-clusters, each containing quarter of the points.

(d) After adding six points to the dataset, the centers stayed stable. Although the radii changed, the micro-clusters are still touching.

Fig. 3 A one-dimensional dataset of 16 points is clustered. Then, six points are added. The larger micro-clusters are affected such that they disconnect the main cluster. The smaller micro-clusters do not disconnect

Fig. 4 Depending on the actual distribution of points, a DenStream instance using smaller micro-clusters can sometimes connect point sets, which are segmented by larger micro-clusters



(a) Micro-clusters in $DS_{0.25}$

(b) Micro-clusters in $DS_{0.3}$

Algorithm 3: AMTICS.merge

```
Data: Two nodes $n_1, n_2$ to be merged
Result: void
Append all children of $n_2$ to $n_1$
Move all weight from $n_2$ to $n_1$
Move all micro-clusters from $n_2$ to $n_1$
Change link from $n_2.parent$ pointing to $n_2$ to $n_1$
Recalculate distances of $n_1$ and potential parents
if $EMD(n_1.parent, n_1) \approx EMD(n_2.parent, n_1)$ then
    $merge(n_1.parent, n_2.parent)$
end
Remove $n_2$
```

described before, we construct a hierarchy and the figure shows both levels and the established micro-cluster structure. As the larger ε value enables a micro-cluster to cover the first three points but not the fourth point, only three points are covered in the final clustering and the remaining point is treated as noise. In the level below, the smaller neighborhood range allows only the first two points to be merged into one cluster. The third point is then processed and cannot be merged into the first micro-cluster. It establishes a second micro-cluster which starts as an outlier micro-cluster. Then, it is merged with the remaining point, which allows this cluster to be raised into a potential micro-cluster. The alignment step will align the one cluster in the top level to all clusters in the bottom level, causing the larger micro-cluster to cover less points.

To repair the hierarchical structure and induce the monotonicity required for a reachability plot, we virtually add cluster points to the parent clusters such that their weight exceeds or equals the weight of the children clusters. Our assumption here relies on the better coverage quality of smaller micro-clusters for arbitrarily shaped clusters. Technically, the hierarchy has to be processed bottom-up and for each parent cluster the weight sum of all directly related is calculated. If this sum exceeds the parent cluster weight, its weight is set to the sum. Otherwise, nothing has to be done. Eventually, the root DS_∞ is reached. It is very common that the weight sum of the second level will exceed the weight of the top cluster. However, this cluster covers by definition all points of the dataset. The ratio of the number of points that it

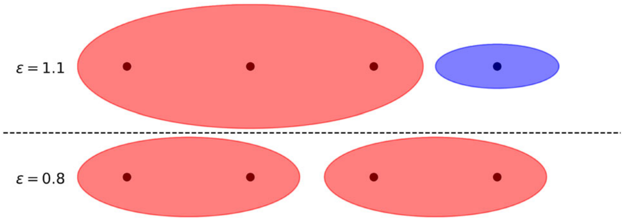


Fig. 5 Clusters of DenStream instances with higher ε values can split into clusters with a higher point coverage due to processing order and the micro-cluster architecture

should have to the number of points it would have regarding the weights of its children clusters is then propagated down the hierarchy and used as a scaling factor for all weights.

After this operation, the ε levels present a valid hierarchy ensuring monotonicity between weights of consecutive levels: For all $0 \leq \varepsilon < \varepsilon' \leq \infty$, if a pair of points p_1, p_2 is clustered in the same cluster in DS_ε, then p_1 and p_2 are also clustered in one cluster in $\mathrm{DS}_{\varepsilon'}$.

3.5 Generating the Reachability Plot

To yield a reachability plot, we construct a dendrogram of horizontal bars B in a first step. We approach top-down through the hierarchy and transform the hierarchy of clusters into a tree of line objects, where each line represents a cluster. For each cluster, we plot a horizontal bar $b = (x, y, w) \in B$ with (x, y) being the starting point and w being the bar width. The width is defined by the weight sum of all contained micro-clusters and represents the cluster size. For the height, we define $y = \varepsilon$. The first bar for DS_∞ will be drawn with $y_0 = 1.25 \cdot \max_{\varepsilon \in \mathbb{R}}$ as $b_0 = (0, y_0, \mathrm{DS}_\infty.weight) \in B$.

Recursively, if a bar has been drawn as $b = (x, \varepsilon, w)$ and the according cluster has n children clusters $c_1, \ldots, c_n$, we first compute the remaining space by $\mathrm{rem} = w - \sum_{i=1}^{n} c_i.weight$. All children bars are distributed equally, using rem/n as an intermediate space between them. As we ensured in the previous section that the sum of all children weights will not be larger than the parent weight, the intermediate space will be zero at least.

The final refinement is the definition of the reachability $r(z) = \min\{y > 0 \mid (x, y, w) \in B \wedge x \leq z \leq x + w\}$. Geometrically, we sweep-line from left to right and choose always the lowest bar of all candidates at this point on the x-axis. As b_∞ spans the complete interval, a minimum can always be found.

3.6 Limitations and Complexity

The used Earth Mover's Distance is quite slow in performance. However, the overall complexity is not changed as we are only looking on finitely many DenStream instances. The aim of the EMD is to distribute one set of objects onto a set of bins with the same capacity. In our case, we explicitly use it as a distance between differently sized sets. Since the distance computation is a core step in our method, it might be worthwhile to improve this step further, for example by introducing a suitable index structure to reduce the number of distance computations or lower distance bounds for candidate filtering. However, alternatives to EMD and more in-depth evaluations on other distance measures are not in the scope of this paper and will be addressed in future works.

A strong limitation and simultaneously a benefit is the possibility to initialize and remove ε instances at any time and for any density level. It is quite difficult to choose the first few instances in the case of the absence of all expert knowledge over the dataset. After some key levels have been identified, the interactiveness is quite useful to get a more accurate picture for certain density levels. Giving a reasonable start environment depending on the fed data automatically would improve the usability significantly.

Regarding the complexity, each DenStream instance keeps at most W/MinPts many potential micro-clusters in memory as shown in [3]. As W is the overall weight of all clusters, it can be replaced by $W = v/(1-2^{-\lambda})$ with v being the number of objects observed in the stream per time unit. Since we keep k instances of DenStream in parallel, the complexity is given by $\mathcal{O}(kv/(1-2^{-\lambda}))$ or $\mathcal{O}(k/(1-2^{-\lambda}))$ per stream object. As we only introduce finitely many clustering instances, the complexity is constant for $\lambda > 0$. Solving optimizations like Earth Mover‘s distances is expensive, however, as the number of micro-clusters is finite, the hierarchy is constructed in constant time. This also holds for the number of hierarchy nodes. Although it is not required for the final result, we can ensure a constant complexity for the whole chain of operations.

3.7 Initial Parameter Choice

In the beginning, one starts with exactly two ε-levels ε_0 and ε_∞. However, choosing additional ε-levels to observe the data stream can be difficult. Until suitable levels are identified by trial, interesting clusters could have been missed during the process of randomly probing. Here, we propose a more targeted method to find a good starting set of initial observation levels. Therefore, we collect stream objects in the starting phase and establish a promising set of observation levels in an initial offline phase.

For a predefined $n \in \mathbb{N}$, we apply OPTICS to the subset of n objects. So although we do not apply our novel approach directly from the beginning, we do not lose all information since the offline application of a clustering algorithm still provides us with a result.

Besides identifying interesting structures in the beginning, the OPTICS plot is then used to hint good starting levels

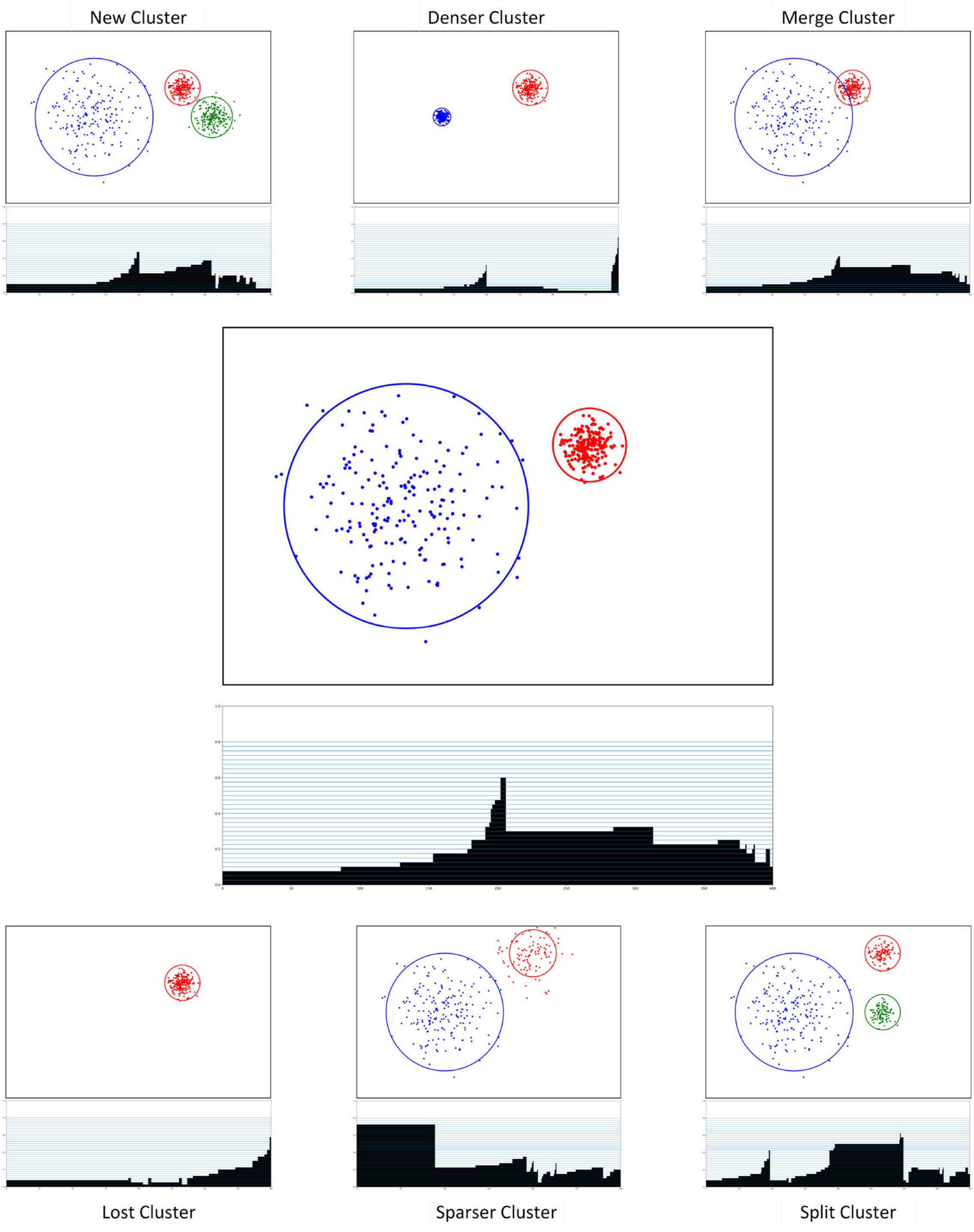


Fig. 6 Influence and observation patterns on AMTICS due to different types of concept drifts. The middle area shows the original pattern before the drift. From top-left to bottom-right, we observe (1) creation of a new cluster, (2) condensing of a cluster, (3) merging of clusters, (4) vanishing of a cluster, (5) extending of a cluster and (6) splitting of a cluster

for our AMTICS. We apply a sweep-line search on the plot horizontally bottom-up. The procedure works in both directions. We are starting at $\varepsilon = 0$. Obviously, the OPTICS plot should not be intersected by our sweep line in the beginning, although it could be tangent in some special circumstances if some points exist directly onto each other. Moving the sweep line up, the number of intersections grows. Everytime the number of intersections changes, the height of the sweep line is used to initialize a new ε-level on this height. Finally, the number of intersections will decrease again until the sweep line is completely above the reachability plot. Then, we do not need to introduce another observation level between the previous one and the level at ε_∞. Choosing those initial observation levels provides us with a good starting point to catch the clustering structure in the beginning of the stream. Next, we will discuss how to maintain an overview about a potential changing structure.

3.8 Concept Drifts and Maintaining the Structural Overview

Observing a stream of objects and their clustering structure, the environment is usually not static. The structure will change and we observe effects that are commonly known as concept drifts. The literature divides between four types of drifts.

Sudden drifts are caused by abrupt changes of the environment. With reference to clustered objects, a sudden drift is often the appearance or vanishing of whole clusters. Due to the fading characteristic of DenStream's micro-clusters, a sudden drift is hardly observable in the AMTICS plot. The decay of a micro-cluster causes a vanished cluster causes a micro-cluster to decay slowly, until the potential cluster components traverse to the outlier state and become ignored in the result set.

However, sudden drifts are very rare in real-world applications and often caused by active and obvious manipulations like switching a lever. In many cases, sudden drifts are incremental drifts with very small transition times. In the case of incremental drifts, a cluster appears or vanishes over some time. This effect can be seen as troughs in the plot change their shape. For a vanishing cluster, the shape becomes smaller and more shallow and an emerging cluster leads to a valley, growing in size. In Fig. 6, starting from the central dataset in the middle, the left top area shows the dataset with the corresponding AMTICS plot after another cluster has been emerged. Instead of two dents in the plot, three troughs are observed, according to the changed dataset. In the left bottom area of the figure, the large cluster has been erased. The AMTICS plot correctly shows only one large trough since there is only one cluster in the data left.

The existence of clusters is not the only factor that can change dynamically. The density is a dynamic property as well. The number of objects in a cluster might be unchanged, but if the spatial variance increases, the density will decrease. In this scenario, the corresponding trough will flatten over time, while keeping the same diameter. Analogously, the same holds for clusters that converge toward their center. Increasing density causes dents in the plot to be deepened. The ability to recognize such a density drift with AMTICS is highly depending on the resolution of density levels and the significance of the density change. If the drift has only minor significance, the drift remains undetected if the gap between two density levels is too large. In Fig. 6, the middle top shows the results after the larger cluster has been condensed into a smaller area, although containing the same number of objects. The corresponding trough is deeper now, indicating a denser connectivity within the cluster. The separating peak still highlights the structure comprising of two distinct clusters. On the opposite position in the middle bottom section, the smaller cluster has been extended. In the AMTICS plot, we can clearly observe that the levels of both dents are similar now, as in the case of increased density before. However, the separating peak has almost vanished. This is also explainable, since border points of both clusters are already mingled. The plot also shows a black area of points that could not be assigned to a cluster. The DenStream instances at these levels produced a number of outlier micro-clusters. According to the scatterplot, this result is expectable since many points exist in sparse border areas now.

Spatial movement is also a drift effect, which is caused by changing attributes. The position is not tracked in the reachability plot; hence, such a drift cannot be detected by AMTICS. However, this is not a disadvantage, since the main focus is on the clustering structure. A side effect of moving clusters is merging and splitting of clusters when clusters overlap. This effect is detected by AMTICS as two closely neighboring clusters will eventually merge their troughs in the plot when they meet in the data space and a kind of interference pattern will indicate a joint dense cluster. In Fig. 6, the right sections show such operations. In the top right corner, the denser cluster collides with the larger cluster. The result in the AMTICS plot is observed as the separating peak is shrinking in height. If both clusters converge toward a common center, the peak will disappear and there will be one large dent with an accumulated depth. In the bottom right corner, the denser cluster has been split into two parts. In the corresponding AMTICS plot, we can differentiate between three clusters. Some objects of the sparse large cluster on the left are very close to one of the new clusters, which causes the lower DenStream instances to apply an unstable cluster assignment. This is a common problem of density-based clustering when cluster densities are too different and intercluster distances are too small.

Sometimes, changed circumstances only affect a subset of objects while the other part of objects follows the for-

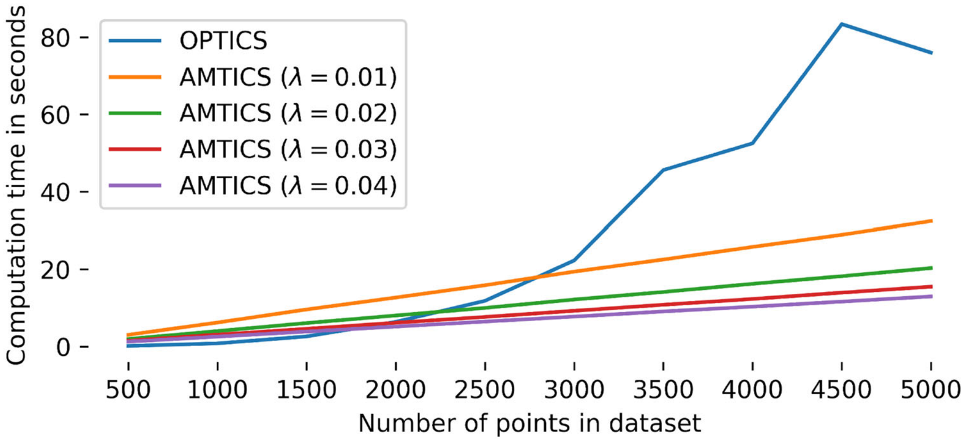


Fig. 7 Comparison of computation time for OPTICS and different AMTICS instances with varying decay factors. While OPTICS has a quadratic complexity, AMTICS shows its linear growth for increasing numbers of data points

mer behavior. For example, there might be contracts that assure some objects former rights, leading to different observations. In a certain period, objects with a changed behavior will appear while the old behavior remains dominant at first. During this period, the new behavior eventually dominates the former behavior until objects with the former behavior will disappear completely. In such a case, we are investigating gradual concept drifts. However, to identify such drifts, we need a higher level of observation perspective and it is required to store and compare different snapshots of the clustering structure. Although AMTICS might provide those snapshots, the analysis of such drifts is a topic of its own and not covered here.

Also, concept drifts are not necessarily restricted to unique occurrences. If we have some kind of seasonal impact on our data and drifts will occur regularly, for instances the system will oscillate between two behaviors, we are talking about a recurring drifts. The same as for the gradual drift holds here, since the identification involves more stream observation techniques and is therefore also not covered in this work.

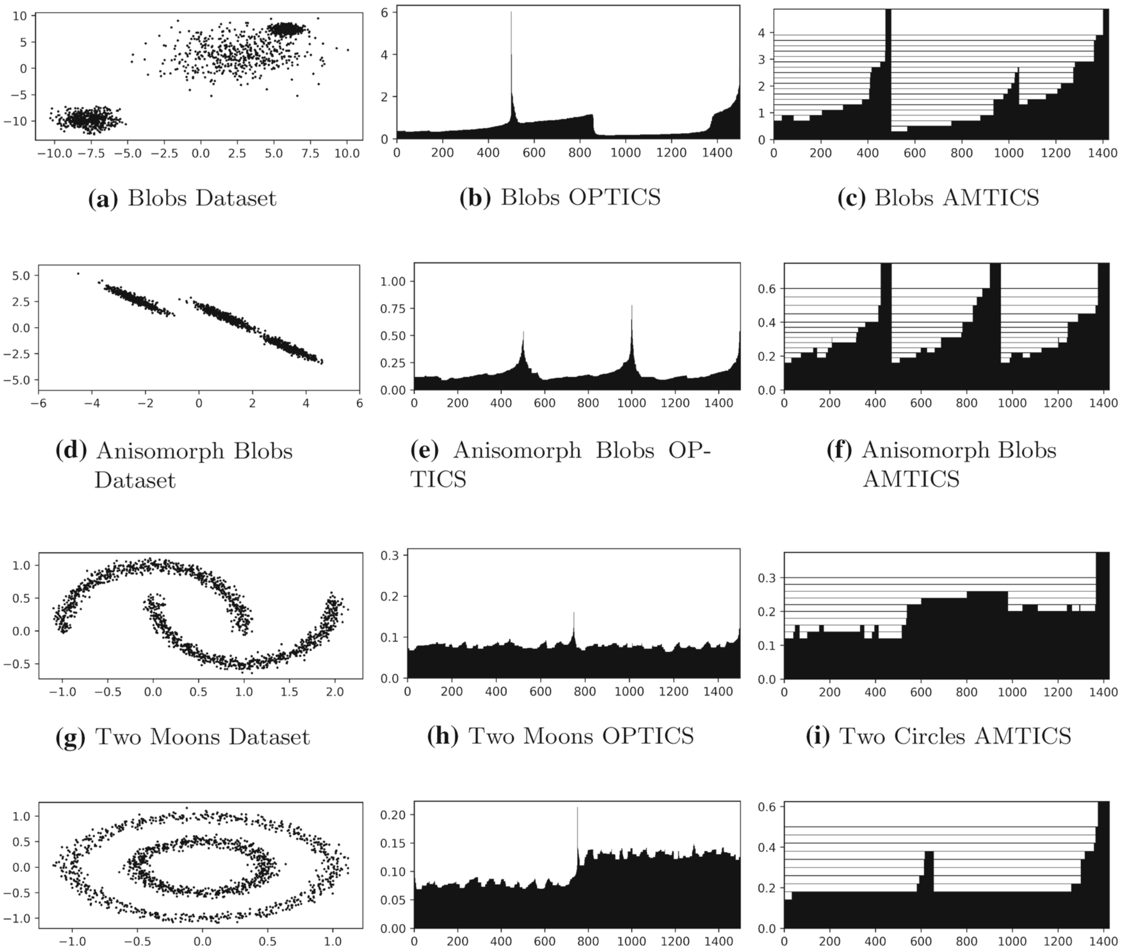
4 Evaluation

AMTICS is a data exploration tool to get first results of the clustering structure within a data stream. To be applicable to streams, an online algorithm has to process each object in $\mathcal{O}(1)$. We empirically prove this claim to be correct by measuring the performance on a data stream. The two moons dataset was already mentioned in the previous section. We generated two moons datasets with various sample sizes between 500 and 5000 points. As a baseline, we applied OPTICS to these datasets and measured the computation time. We repeated each computation three times and used the minimum to compare with our algorithm. For AMTICS, we used four decaying factors $\lambda \in \{0.01, 0.02, 0.03 and 0.04\}$ as we know that the decaying influences the number of micro-clusters and thus the performance. To get consistent results, we applied AMTICS with the same settings on the streamed datasets repeatedly. The complete computation time over all stream data points is aggregated. For all AMTICS runs, we used five ε instances.

In Fig. 7, we plot the computation time in seconds for the different dataset sizes. All evaluations were performed on a workstation with an Intel Xeon CPU with 3.10 GHz clock frequency on 16 GB memory. The results show a clear linear increase in computation time for AMTICS which is expected since the additional processing time for each arriving item has constant complexity. In comparison with OPTICS which shows a quadratic complexity in the number of data points, AMTICS has a reliable linear complexity over the number of data items. Obviously, OPTICS can be faster for small datasets where point neighborhoods consist only of few points. However, already for medium-sized datasets, the computation time of OPTICS exceeds our efficient method.

To achieve the performance advantage, we trade time for accuracy. As we only produce an approximation of the OPTICS plot, the result can be coarse. This is especially true in the beginning of a stream and for a small set of ε instances. To get an impression of how AMTICS results look like compared to OPTICS and to compare the detected clusters, we refer to some 2d example datasets in Fig. 8 and the popular chameleon dataset in Fig. 9. In the four test cases, AMTICS can compete with OPTICS by identifying the overall structure of the clusters, although the OPTICS results are slightly more accurate. The most difficult scenario for AMTICS seems to be the two-circle dataset. Due to the curving, the micro-clusters in the inner circle are touching the outer circles rather early. This leads to a rather large transition phase between the detection of two clusters and one cluster only, which can be seen in the corresponding AMTICS plot.

To give AMTICS a little challenge, we also clustered the chameleon dataset, which is tough due to the combination of solid clusters, sinusoidally arranged points and much noise. Both OPTICS and AMTICS struggle with the detection of the solid clusters. However, AMTICS is still able to identify



(a) Blobs Dataset

(b) Blobs OPTICS

(c) Blobs AMTICS

(d) Anisomorph Blobs Dataset

(e) Anisomorph Blobs OPTICS

(f) Anisomorph Blobs AMTICS

(g) Two Moons Dataset

(h) Two Moons OPTICS

(i) Two Circles AMTICS

(j) Two Circles Dataset

(k) Two Circles OPTICS

(l) Two Moons AMTICS

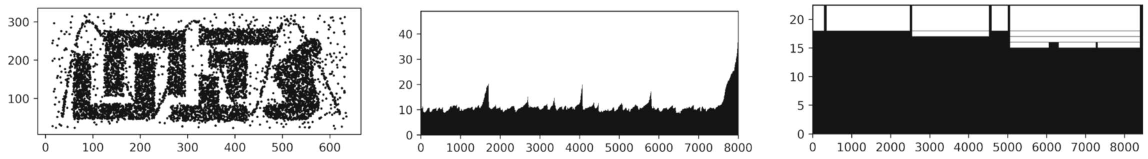
Fig. 8 OPTICS and AMTICS in comparison on different synthetic datasets

the cluster structure with only few ε instances. Some clusters are connected with dense noise, so they are aggregated in the result.

OPTICS and AMTICS use a parameter ε with a synonymous meaning. We investigated the correlation between both values $\varepsilon_{\text{OPTICS}}$ and $\varepsilon_{\text{AMTICS}}$ and define a controllable test scenario 'GridGap' as sketched in Fig. 10. It consists of two regular grid-based point sets with a defined margin d in between. If d is larger than the point distance in one of the grids, it corresponds directly to $\varepsilon_{\text{OPTICS}}$. The maximum ε for AMTICS to divide the dataset is $\varepsilon_{\text{AMTICS}}$. Since instances have to be defined in advance, we choose a step width of 0.02.

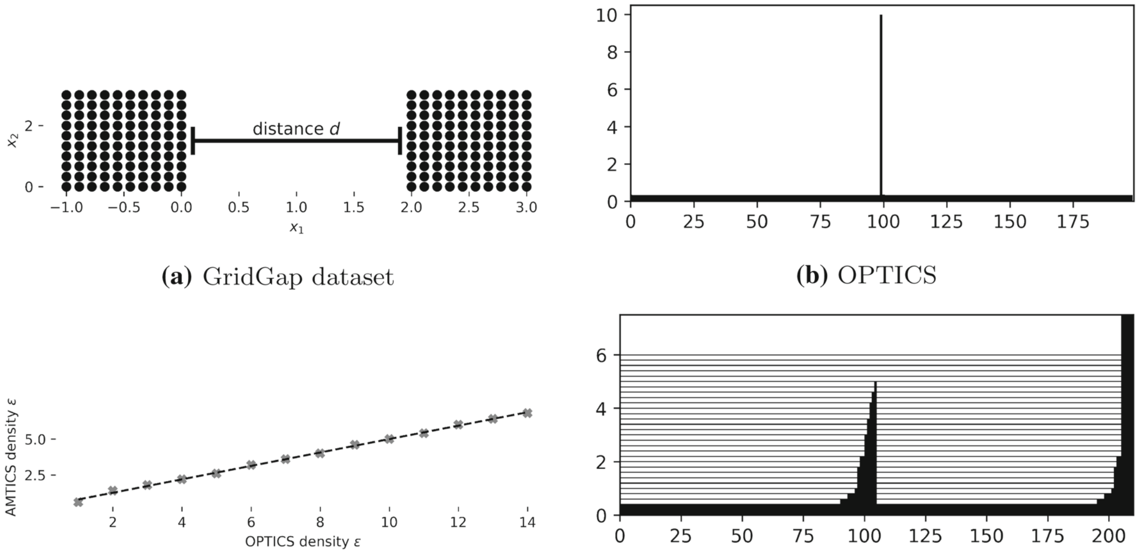
In Fig. 10, we exemplarily gave the resulting plots for $d = 10$. The overall structure is represented. We compared the peak height of the OPTICS plot with the peak height of the needle-like pillar in the middle of the AMTICS plot. Evaluating several distances, we concluded that both approaches are correlated linearly by $f(x) = 0.47 \cdot x + 0.33$. The constant term is a result of the distance within both clusters. In the case of OPTICS, the distance is constant 0.1 for a grid of 10×10 points. For AMTICS, the distance between micro-clusters is used. Since micro-clusters represent a Gaussian distribution but we distributed the points equally, distances are stretched in comparison with OPTICS. Hence, we assume that the actual distribution of the data points will influence this constant offset and it should not be expected that the same density level in both methods yields the same results.

In a last evaluation, we show the applicability of AMTICS to datasets of higher dimensionality. The datasets contain 2000 data points partitioned into three circular clusters of equal size. The numbers of dimensions are 10, 20, 30, 40, 50 and 100. In Fig. 11, we give the AMTICS plots for the datasets. For higher dimensions, the distances rise and obviously the density-based method will fail for really high dimensions due to the curse-of-dimensionality and the fact that differences between distances of object pairs will vanish. For lower dimensions and practical applications, our



(a) Chameleon Dataset **(b)** OPTICS **(c)** AMTICS

Fig. 9 OPTICS and AMTICS in comparison on the Chameleon dataset



(a) GridGap dataset **(b)** OPTICS

(c) Correlation **(d)** AMTICS, 30 instances

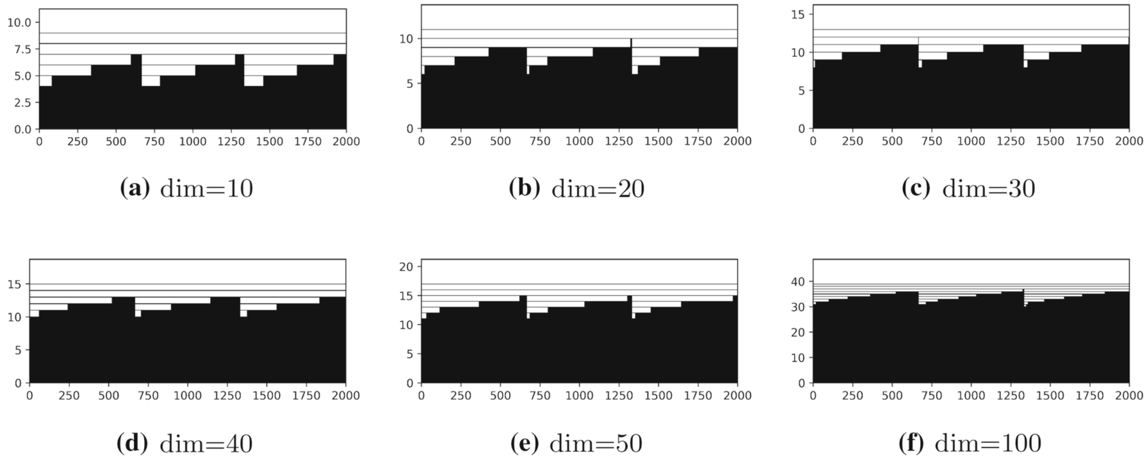
Fig. 10 Evaluation setup to estimate the correlation between density in OPTICS and density in AMTICS. Two regular point grids are divided into two clusters. The gap distance in between is increased incrementally. Comparison of two result plots of OPTICS and AMTICS on the GridGap dataset with $d = 10$

approach is able to distinguish between the three clusters. This test is performed using the Euclidean distance. At some point, it might be useful to substitute the distance with a more robust measure like the Mahalanobis distance. In low-dimensional spaces, the speed advantage of the Euclidean distance prevails.

5 Related Work

In [1], the authors provide a formidable survey paper which encompasses state-of-the-art density-based stream clustering algorithms. To resharpen the focus: AMTICS is a density-based stream clustering algorithm which comes with the following properties: (a) it relies on concepts inherent to OPTICS, (b) enables the operation on multiple resolutions and (c) permits interactivity by interception while at the same time reducing re-computation efforts. Considering the algorithms proposed in [13] which are similar to AMTICS w.r.t. the concepts provided by OPTICS, StreamOptics is listed. While this method satisfies the criterion of relying on the concepts of OPTICS, it neither permits to investigate the clustering at different resolutions nor does it provide interactions from users. Regarding the support of different resolutions, the authors in [1] refer to the method MR-Stream [15]. However, while enabling multiresolution clustering, MR-Stream considers one single and therefore globally applied ε-range and minpts parameter setting, while in contrast OPTICS and AMTICS can identify the different ε-ranges of each cluster. It also does not provide any interactions. AMTICS serves the purpose to inspect interactively different density levels and further investigate the potentially emerging clusters. By selecting interesting regions on different density levels, the information is merged into an approximative OPTICS plot. The fact that AMTICS has a human-in-the-loop component renders it difficult to perform quantitative analysis w.r.t. quality measures such as NMI, ARI or density-based validation index [9]. With the aspect of interactiveness, it has a distinctive property which is not captured by any of the state-of-the-art properties elaborated on in [1].

Since AMTICS is targeted at a density-based clustering model, we elaborate first on DBSCAN [4] as among



(a) dim=10 **(b)** dim=20 **(c)** dim=30

(d) dim=40 **(e)** dim=50 **(f)** dim=100

Fig. 11 AMTICS applied to three-blob datasets with multiple dimensions

the most prominent algorithms for density-based clustering being followed by GDBSCAN [12], HDBSCAN [8] and DENCLUE [6]. In DBSCAN, the users set an ε and minimum number of objects to be located within this range by which the expected density is characterized. Nevertheless, DBSCAN also comes with certain weaknesses. One of them is that it is a challenging task to determine an adequate ε-range. A second weakness is its incapability to detect clusters of different density. A method which was constructed as a visualization and has been constructed with these two weak points in mind is OPTICS [2]. To overcome these weak points of DBSCAN, OPTICS orders the points of a given dataset in a linear fashion, such that neighboring points in this ordering are following consecutively after each other. In the form of a reachability plot, users can spot valleys of different depths by which (1) the different densities of clusters become visible and (2) determining an adequate ε-range is facilitated.

Data can occur in a stream setting, DBSCAN and OPTICS are not suitable for high-velocity scenarios. As such, a density-based clustering algorithm tailored at data streams has been designed known as DenStream [3]. Here, the authors introduce dense micro-clusters (core-micro-clusters) with the purpose to summarize dense clusters of arbitrary shapes. Having a density-based method for the stream setting, works have emerged on providing an OPTICS-fashioned method for high-velocity scenarios. As one related work, we have OpticsStream [13] which hybridizes the concept of a density-based stream clustering with an extension of OPTICS. Their approach to a streaming version maintains, based on the reachability distance, an ordered list of core micro-clusters. The maintenance over time is ensured through insertions and deletions in a separate micro-cluster list. As a result, an OPTICS plot is generated based on the current micro-cluster structure. Further, the authors introduce a three-dimensional reachability plot where the third dimension is an axis representing the time. However, the three-dimensional construct renders it difficult to clearly identify the valleys and thus different density levels. One of the major differences between OpticsStream and AMTICS is that our approach does not start one single instance of an density-based stream clustering but several. This gives the opportunity to detect clusters even though they are changing their density over time.

For the OPTICS algorithm itself, the authors recognized in IncOPTICS [7] that it is not necessary to compute the whole reachability plot anew. By the notion of density in OPTICS, insertions and deletions impact only on a small subset of objects. Updates, however, affect another subset of objects leading to movements within the clustering order. Based on these observations, the authors provide in their work an algorithm to incrementally insert and delete within a cluster ordering. However, the two major steps are infeasible w.r.t a streaming setting. The first step involves the detection of a starting point for a reorganization which is required after each update operation. The second step involves the reorganization of the cluster ordering until a valid ordering is re-obtained.

An OPTICS variant which is especially tailored at large volume data is GridOPTICS [14]. The fundamental idea behind this work is to impose a grid on a given dataset. The grid approach the authors use is not the area or hypervolume of a grid cell per se but the junctions, or in a more illustrative way: the corner points of a grid cell. The objects are assigned to their closest junction point yielding junction subsets. In the following step, it executes OPTICS on each of the grid junction subsets. Then, in the third step, the clusters per junction subsets are derived from the OPTICS computation. Finally, all objects over all junction subsets are assigned to their, respectively, closest cluster which leads either to no changes of junction subsets or to a merging of them. While the authors state that this method is in orders of magnitude faster than OPTICS and highly suitable for large volume data, they also state on the contrary that GridOPTICS results can be of lower accuracy compared to OPTICS. Further, the benefits of the runtime can heavily degrade with increasing dimensionality which results in an exponential increase of junctions. AMTICS in contrast is not affected w.r.t. the runtime by increasing number of dimensions as GridOPTICS.

Regarding the aspect of parallelization, POPTICS [10] is a method which is a massive parallel shared memory and distributed memory variant of OPTICS using Prim's Minimum Spanning Tree method. While in POPTICS a low-level approach and inter-process communication management needs to be considered, AMTICS can be parallelized in an very simple fashion on a high-level scale.

In a more recent contribution [5], the authors propose an algorithm which is capable of computing the OPTICS visualizations within $\mathcal{O}(n \log n)$ runtime. This improvement in runtime is at the same time ensured within a certain approximation bound, providing a guarantee that the resulting plots have a highly close resemblance of the original OPTICS plots. In order to achieve this goal, the authors propose a novel approach named ρ-approximate OPTICS. However, the authors state that the bounded precision is ensured in low-dimensional spaces, rendering it infeasible for high-dimensional scenarios.

6 Conclusion

AMTICS is an efficient and interactive density-based online micro-clustering algorithm. It follows a divide-and-conquer strategy by clustering the same dataset on different density levels and merges the separate results into a hierarchy of clusters of various sizes and densities. The hierarchy is finally displayed visually as a reachability plot in which valleys refer to dense areas that are more likely to be clusters.

With AMTICS, it is possible to process data streams and explore the clustering structure visually. It provides the flexibility to shift the focus to certain density levels by incrementally adding or removing clustering instances. AMTICS produces an approximative reachability plot anytime in the stream on demand. Similar to OPTICS, the alignments of micro-clusters provide insights into the cluster structure while in contrast providing a coarse overview for a rapid visual analysis. The advantage is not only the agile construction of a reachability plot, which can be constructed with related methods like OpticsStream [13]. Using the layered hierarchy provides the desired level of cluster granularity which augments the analysis performance as well by reducing the complexity of the visual analysis for humans.

In future works, we are going to investigate heuristics for a useful set of starting ε levels. This will assist a human operator in the visual analysis process. As a further future topic, the distance computation can be improved. Although the Earth Mover's distance is very suitable for the application, we do not need to find the exact distance values. If we provide a sufficient ordering of super-clusters as potential parent nodes in the hierarchy, we can improve the performance in the offline phase.

As mentioned in Introduction, AMTICS does not provide a particular clustering for a dataset. The next logical step in the explorative analysis demands the choice of a clustering paradigm and suitable parameter settings. Hence, succeeding clustering steps need detailed fundamental work and evaluation, which cannot be covered here, although it is a very promising future work.

References

1. Amini A, Wah TY, Saboohi H (2014) On density-based data streams clustering algorithms: a survey. J Comput Sci Technol 29(1):116–141
2. Ankerst M, Breunig MM, Kriegel HP, Sander J (1999) Optics: ordering points to identify the clustering structure. ACM Sigmod Rec 28(2):49–60
3. Cao F, Ester M, Qian W, Zhou A (2006) Density-based clustering over an evolving data stream with noise. In: Proceedings of the 2006 SIAM international conference on data mining. SIAM, pp 328–339
4. Ester M, Kriegel HP, Sander J, Xu X et al (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. KDD 96(34):226–231
5. Gan J, Tao Y (2018) Fast Euclidean optics with bounded precision in low dimensional space. In: Proceedings of the 2018 international conference on management of data, pp 1067–1082
6. Hinneburg A, Keim DA et al (1998) An efficient approach to clustering in large multimedia databases with noise. KDD 98:58–65
7. Kriegel HP, Kröoger P, Gotlibovich I (2003) Incremental optics: efficient computation of updates in a hierarchical cluster ordering. In: International conference on data warehousing and knowledge discovery. Springer, pp 224–233
8. McInnes L, Healy J, Astels S (2017) HDBSCAN: hierarchical density based clustering. J Open Source Softw 2(11):205
9. Moulavi D, Jaskowiak PA, Campello RJ, Zimek A, Sander J (2014) Density-based clustering validation. In: Proceedings of the 2014 SIAM international conference on data mining. SIAM, pp 839–847
10. Patwary MA, Palsetia D, Agrawal A, Liao Wk, Manne F, Choudhary A (2013) Scalable parallel optics data clustering using graph algorithmic techniques. In: Proceedings of the international con-

ference on high performance computing, networking, storage and analysis. ACM, p 49

11. Rubner Y, Tomasi C, Guibas LJ (2000) The earth mover's distance as a metric for image retrieval. Int J Comput Vis 40(2):99–121
12. Sander J, Ester M, Kriegel HP, Xu X (1998) Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications. Data Min Knowl Discov 2(2):169–194
13. Tasoulis DK, Ross G, Adams NM (2007) Visualising the cluster structure of data streams. In: International symposium on intelligent data analysis. Springer, pp 81–92
14. Vagner A (2016) The gridoptics clustering algorithm. Intell Data Anal 20(5):1061–1084
15. Wan L, Ng WK, Dang XH, Yu PS, Zhang K (2009) Density-based clustering of data streams at multiple resolutions. ACM Trans Knowl Discov Data (TKDD) 3(3):1–28