2022 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-5099-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDM54844.2022.00108

DBHD: Density-based clustering for highly varying density

Walid Durani[1], Dominik Mautz[1], Claudia Plant[2,3], Christian Böhm[2]
[1]*Institute of Informatics, Ludwig Maximilian University of Munich, Munich, Germany*
$\{durani, mautz\}$*@dbs.ifi.lmu.de*
[2]*Faculty of Computer Science, University of Vienna, Vienna, Austria*
[3]*ds:UniVie, Vienna, Austria*
$\{claudia.plant, christian.boehm\}$*@univie.ac.at*

***Abstract*—A major challenge in cluster analysis is the discovery of clusters with widely varying sizes, densities, and shapes. Most clustering algorithms lack the ability to detect heterogeneous clusters that differ greatly in all three properties simultaneously. In this work, we propose the Density Clustering for Highly varying Density algorithm (DBHD). DBHD uses a novel approach that considers local density information and introduces two new conditions to distinguish between different types of data points. Based on this and the adaptively computed density information, DBHD can detect the clusters described above and is robust to noise. Moreover, DBHD has intuitive and robust parameters. In extensive experiments, we show that our technique is considerably more effective in detecting clusters of different shapes, sizes, and densities than well-known (DBSCAN or OPTICS) and recently proposed algorithms such as DPC, SNN-DPC, or LSDBC.**

***Index Terms*—Clustering, Density-based, Cluster Analysis, Local Density**

I. Introduction

Most clustering algorithms have strong underlying assumptions about the homogeneity of a cluster's shape, density, or size. In consequence, they perform poorly if a dataset violates one or more of these assumptions. In this paper, we propose DBHD, a method that deals with situations in which all three properties can differ vastly from cluster to cluster. Figure 1 illustrates such a situation: the data contains a very dense area with a large number of clusters that we can only visually see when we enlarge the high-density area considerably. The clusters in the figures are characterized by extreme discrepancies in their densities, shapes, and sizes.

If one tries to identify the clusters of the dataset described above using state-of-the-art clustering algorithms (k-means [1], DBSCAN [2], OPTICS [3] clusters selected via minimum steepness, HDBSCAN [4], [5], DPC [6], SNN-DPC [7], LSDBC [8] Spectral Clustering [9], SpectralACL [10], DBADV [11]), most of the methods provide insufficient results even with the best parameter settings. Only two clustering algorithms can achieve satisfactory results, while the rest achieve weak results.

On average, the methods achieve an Adjusted Mutual Information (AMI) value of only 0.276. Hereby, the best outcome is found by DBADV with an AMI of 0.947, which is better than the average value. However, there is still room for improvement.

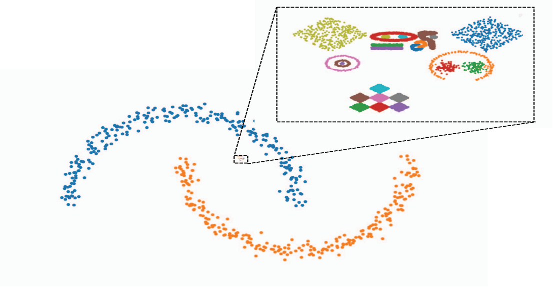

Fig. 1: The diagram shows a toy dataset in which clusters of various shapes with extremely different densities and sizes exist. In this scenario, DBHD achieves an AMI score of 0.990.

In this paper, we present DBHD, a new density-based clustering technique that can detect clusters with strongly varying properties like extremely different scaling.

In summary, we make the following contributions:

- We introduce DBHD, a novel approach to estimate the local density for each cluster and two new intuitive conditions for the assignment of data points to clusters that makes it possible to identify clusters differing extremely in density, size, shape, and scaling.
- Our new approach provides robust and stable clustering results for a wide range of parameters compared to classical density-based approaches like DBSCAN or OPTICS.
- The experimental results demonstrate that our proposed algorithm outperforms state-of-the-art-clustering algorithms while simultaneously being more robust against outliers.

II. Related Work

The most commonly used clustering algorithms can be divided into the following five categories: density-based methods, partitional methods, hierarchical methods, spectral-based methods, and graph-based methods. In density-based methods [12], a dense collection of data points in the feature space

978-1-6654-5099-7/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDM54844.2022.00108

forms a cluster, separated by less dense areas. Each density-based clustering algorithm defines a unique concept of density. Based on this notion, a density-based clustering algorithm can find clusters of arbitrary sizes and shapes. Another advantage of this approach is its robustness against outliers and that the users do not have to determine the number of clusters. The best-known density-based algorithm is DBSCAN [2], which defines a global density threshold through the defined parameters. This allows DBSCAN to find arbitrarily shaped clusters, but it makes it also impossible for DBSCAN to detect clusters with varying densities. A wide variety of density-based clustering algorithms have been published based on DBSCAN , providing different approaches for detecting clusters with different densities (OPTICS [3], HDBSCAN [4]. In recent years, several density-based methods have also been published that exploit specific properties such as local density or density peaks to detect clusters with different densities. LSDBC [8] DPC [6] SNN-DPC [7], DBADV [11]

Partitional and spectral clustering methods also have high relevance in practice and are frequently used. One of the best-known partitioning-based clustering algorithms is the *k*-means clustering algorithm. In the first step, k cluster centers are initialized randomly. Then, in an alternating process, the data objects are assigned to the nearest cluster center, and the new cluster centers are calculated. There are numerous extensions to *k*-means that tackle various data mining problems. For example, the SubKmeans and Nrkmeans [13], [14] extensions to *k*-means can identify a subspace where the clustering are located and multiple non-redundant clustering hiding in different subspaces of the dataset. The spectral clustering [9] algorithm has been widely used in real applications. This approach creates a similarity graph of the data and transforms it into the Laplacian matrix. Based on this, it calculates the k eigenvectors corresponding to the smallest eigenvalue. In the last step, *k*-means is applied to the entries of the eigenvalues. There are countless algorithms based on spectral clustering. One of the most interesting one is SpectACL [10], which attempt to overcome spectral clustering's weaknesses by combining spectral clustering with density-based clustering. SpectACL determines the density for all clusters over the spectrum of the weighted adjacency matrix in the data. Therefore, it is able to discover clusters with varying densities. Mean-shift is a famous centroid-based clustering algorithm, which aims to find the mode of a density function. In the first step, it randomly initializes a data point as a mode and iteratively updates the mode until a convergence criterion is fulfilled [15].

III. DBHD – Local Density Clustering

In the following, we describe the DBHD algorithm. First, we lay the ground with some basic assumptions and necessary definitions. Next, we provide a high-level perspective onto the algorithm, and we introduce the general idea behind DBHD. Finally, we describe the algorithm. An implementation of DBHD and for the experiments can be found under https://dm.cs.univie.ac.at/research/downloads/.

A. Underlying Assumptions and Basic Definitions

We are given a dataset $\mathcal{D}$ of objects for which we want to find the structures of the cluster. To find meaningful structures within the dataset, we need a way to measure $\operatorname{dist}(\cdot,\cdot)$, the distance between two data points.

Our goal is to find arbitrarily shaped clusters with varying numbers of data points that also vary widely in density. The first basic assumption of DBHD is that a cluster contains at least *n* objects - the cluster's minimum size. This is the first of three intuitive parameters that a user must specify.

Since we want to find clusters of varying density, we first have to define how we measure this density. It is quite common for density-based clustering algorithms to determine the density of a region around a point via a concept called the ϵ-neighborhood:

Definition 1 (ϵ-neighborhood of a point x): We define the neighbors of a point $x \in \mathcal{D}$ by a given ϵ as the set:

$$\mathcal{H}(\mathcal{D}, x, \epsilon) = \{y \in \mathcal{D} \mid \operatorname{dist}(x, y) \leq \epsilon\}. \qquad (1)$$

That is, given a data point x of the dataset $\mathcal{D}$, there is a ϵ-neighborhood containing all objects for which the distance to x is less than or equal to ϵ. Here ϵ is often a parameter that the user must specify. The density of a region around a point is then estimated by the number of data points in its ϵ neighborhood. However, in DBHD, we assume that density can vary greatly between clusters, and therefore, one cannot use this concept directly.

Instead, we have to assume that the density of clusters varies within the dataset. We use this variation to introduce the region's property with the highest density around a point: Given the variation, there exists a cluster with the highest density. Of course, each cluster's density may also vary slightly (how much variation DBHD allows is a user-specific parameter that we introduce below). As a result of this variation in density within the cluster, this cluster also contains a region around a point σ with the highest density.Before we continue, we should note that in general, there might be multiple clusters with multiple regions with an equally greatest density. In the extreme case, all clusters share the same density without any variation in their density. In such a case, we will select the point, which has the smallest value along one dimension as σ (see the reproducibility section for more details), and DBHD will behave like a modified version of DBSCAN. For brevity's sake, we assume in the following that there exists only one point of greatest density σ. The central underlying idea of DBHD is to start extracting cluster structures sequentially, beginning by the cluster of greatest density and finishing at the least dense cluster. Because of this sequential approach, we must distinguish between the entire dataset $\mathcal{D}$ and its subset $\mathcal{D}^*$ of objects that we have not yet assigned to a cluster.

Since we assume that a cluster contains at least *n* objects, we can exploit the average distance of a point x to its *n*-nearest neighbors as an estimate for the density of region around this point. And the point of greatest density σ is the point with the least average distance to its *n*-nearest neighbors. The following three definitions formalize this concept.

Definition 2 (n-nearest neighbors): For a data point $x \in \mathcal{D}^*$ and a given minimum cluster size n, we define the n closest (according to $\text{dist}(\cdot,\cdot)$) unprocessed neighbors $y_1, \dots, y_n \in \mathcal{D}^*$ as the set:

$$\mathcal{Q}(x) = \{y_1, \dots, y_n\}. \tag{2}$$

The size $\mathcal{Q}(x)$ is always n. If more than n data points satisfy the n-nearest neighbors property for x, one select the first n-nearest neighbors as $\mathcal{Q}(x)$.

Definition 3 (average neighborhood distance): The average distance of a data point x with $\mathcal{Q}(x) = \{y_0.., y_n\}$ is defined as:

$$\text{avg}_{\mathcal{Q}}(x) = \frac{\sum_{y \in \mathcal{Q}(x)} \text{dist}(x, y)}{n} \tag{3}$$

Definition 4 (point of greatest density σ): We define the data point of greatest density σ of $\mathcal{D}^*$ for a given minimum cluster size n as the data point in $\mathcal{D}^*$ with the smallest neighborhood average distance from all data points in $\mathcal{D}^*$. The most important property of σ is that it is part of the dataset's current cluster of greatest density.

$$\sigma = \min_{x \in \mathcal{D}^*} \text{avg}_{\mathcal{Q}}(x) \tag{4}$$

With these definitions, we have a starting point for the extraction of a cluster and can begin to expand it around this area, as explained in the next section.

B. Cluster Expansion

In DBHD, we utilize for the cluster expansion procedure the ϵ-neighborhood as defined above. Instead of letting the user set the ϵ-value, we estimate it from the data at the beginning of each cluster expansion. The ϵ-value is estimated based on the minimum distance over all data points to their n-th neighbor:

Definition 5 (Epsilon ϵ): We define ϵ as:

$$\epsilon = \min_{x \in \mathcal{D}^*} \max_{y \in \mathcal{Q}(x)} \text{dist}(x, y). \tag{5}$$

We expand a cluster by starting at the point of greatest density σ and using the ϵ-neighborhood (definition 1) to expand the cluster in an iterative process. During the expansion, we have two different sets: C contains data points that we have assigned to the *cluster* in the previous iterations of the cluster expansion. The set P contains *potential* new members of the cluster that will be added to the cluster at the end of the current expansion step if they satisfy two conditions: (1) the first condition considers the density variation, (2) the second condition considers for each object the distribution of its neighbors.

Both conditions are described in more detail in the following section. At the end of each expansion step, we replace P with the objects of the ϵ-neighborhoods of the newly added data points. We stop the expansion process once C does not change anymore. While expanding a cluster, we only assign data points from P to the cluster C that satisfy both conditions. If one condition is violated, we reject the data point for this expansion step. Yet, we might reconsider a previously rejected data point in a later iteration. Once the cluster expansion is stopped, we mark all data points in C as processed and start the search for the region of greatest density within the remaining unprocessed data objects.

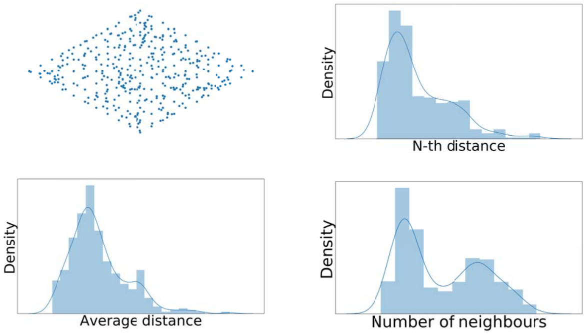

Fig. 2: The figures illustrate how key properties of a data point can vary within a cluster.

When we have extracted all clusters, the remaining data points that cannot be assigned to a cluster are considered as noise and form the outlier set.

C. The Two Membership Conditions

If clusters were homogeneous, all data points would have a very similar density to the σ. In this case, one can easily estimate the cluster's overall density and determine all data points that belong to the cluster.

Unfortunately, clusters are usually inhomogeneous. Especially, the density in a cluster varies to a certain degree. Therefore, it can be challenging to decide whether a data point belongs to the current densest cluster. To illustrate this point, let us consider the example in Figure 2.

Although all data points belong to one cluster, they differ in their values for different properties such as n-th neighbor distance, average neighborhood distance, or the number of neighbors at a given ϵ. Therefore, when deciding whether a data point belongs to a cluster, the fluctuations in the density must be considered. Many approaches simply use the number of neighbors of data points at a given ϵ to estimate if a data point belongs to the current cluster [2] or use only the local density information [8] to measure the membership of data points. All these information are not sufficient to capture the crucial properties of arbitrary shaped clusters.

With DBHD, we propose a novel approach that utilizes two conditions described in the following that must be fulfilled by a data object to be assigned to a cluster. The first condition accounts for the density variation. The second condition takes for each object the distribution of its neighbors into account.

1) 1. Condition – Density Variation: The first condition allows for a slight density variation for data points which we potentially want to add to a cluster. At the same time, it ensures that we do not add local outliers to a cluster. We can utilize that outliers, which are located around a cluster, have a higher average distance to their k-nearest neighbor than points that

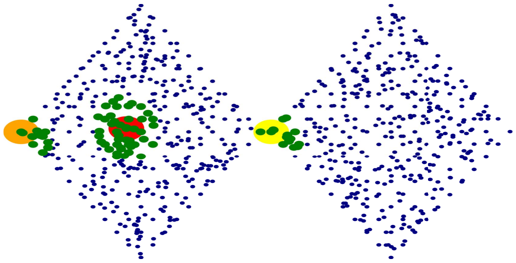

Fig. 3: The diagram shows some exemplary data points, a red central point, an orange border point and a yellow connecting point together with their neighborhoods (highlighted in green). We show a red central point, an orange border point, and a yellow connecting point. The neighborhood of each point is thereby highlighted in green.

are belonging to the cluster. Combining these, we check if for a data point x if its average neighborhood distance exceeds a threshold compared to the average neighborhood distance of σ:

$$\beta \times \mathrm{avg}_{\mathcal{Q}}(x) \leq \mathrm{avg}_{\mathcal{Q}}(\sigma), \tag{6}$$

where β, with $0 < \beta < 1$, is the second user-specified parameter of DBHD. It allows for the variation of the densest region and least dense region within the cluster, and at the same time, it allows to reject local outliers.

2) 2. Condition – Neighborhood Distribution: Based on the first condition, it is challenging to decide for data points located between several clusters to which cluster they belong. As a result, we need a second condition that considers how the n-nearest neighbors of a data point x are distributed between the cluster and its outside surrounding. Therefore, we check how many neighbors of a data point x lie within the cluster C and potential members P compared to how many neighbors lie outside of the cluster and its potential members. Formally, we define the second condition as

$$\rho \times |\mathcal{H}(\mathcal{D}^*, x, \epsilon) \setminus (C \cup P)| \leq |\mathcal{H}(\mathcal{D}^*, x, \epsilon) \cap (C \cup P)|, \tag{7}$$

where we introduce the third and last parameter ρ, which allows for some variations between the two sides. If we set $\rho = 1$, it would mean that there must be at most as many neighbors of x within $C \cup P$ than outside of it. If there are more neighbors outside of $C \cup P$, we reject the data point.

3) Illustration for both Conditions: In the following, we provide a visual explanation of how the previously discussed conditions work in detail. We introduce three new kinds of data points to describe the underlying idea:

- *Central points*: data points that are centrally located in the cluster.
- *Border points*: data points that are the data points that define the border of a cluster.
- *Connecting points*: data points that are located between clusters.

In Figure 3 one can see that the n-nearest neighbors from *central points* are distributed in all directions of the data space. In contrast to data points, which are located at the border of the cluster, the n-nearest neighbors of *border points* (as shown in Figure 3) are spread in a limited way in the data space. Thus, the n-nearest neighbors of border points are further away from *central points*. It follows that the average distance from inside cluster points is smaller than the distance from the border points and especially from the outliers. This is the property, which exploits the first condition. With the first condition's help, we can distinguish between border points and outliers around a cluster. Moreover, Figure 3 shows a *connecting point* whose majority of its n-neighbors are in the right cluster and only some neighbors are in the left cluster. Based on the second condition, this data point is assigned to the correct cluster.

D. Algorithm

Algorithm 1: DBHD

```
input  : dataset: $\mathcal{D}$, minimum cluster size: $n$,
         cluster member ratio: $\rho$, density deviation threshold:
$\beta$
output: clusterSet, outlierSet
// Initialization:
$\mathcal{D}^* := \mathcal{D}$ // The set with all unprocessed points
clusterSet := $\{\}$
// Main Loop:
while $|\mathcal{D}^*| > n$ do
    $\epsilon := \min_{x \in \mathcal{D}^*} \max_{y \in \mathcal{Q}(x)} \mathrm{dist}(x, y)$
    $\sigma := \min_{x \in \mathcal{D}^*} \mathrm{avg}_{\mathcal{Q}}(x)$
    $C := \{\sigma\}$ // The object set for the new cluster
    $P := \mathcal{H}(\mathcal{D}^*, \sigma, \epsilon)$repeat
        $M := C \cup P$ for $x \in P$ do
            $N = \{\}$ // The set of new cluster points
            if $\beta \times \mathrm{avg}_{\mathcal{Q}}(x) \leq \mathrm{avg}_{\mathcal{Q}}(\sigma)$
              $\wedge\ \rho \times |\mathcal{H}(\mathcal{D}^*, x, \epsilon) \setminus M| \leq |\mathcal{H}(\mathcal{D}^*, x, \epsilon) \cap M|$
              then
                $N := N \cup \{x\}$
            end
        end
        $C := C \cup N$
        $P := \left(\bigcup_{x \in N} \mathcal{H}(x, \epsilon)\right) \setminus C$
    until $C$ does not change anymore;
    clusterSet := clusterSet $\cup\ \{C\}$
    $\mathcal{D}^* := \mathcal{D}^*\ \setminus C$
end
outlierSet := $\mathcal{D}^*$
return clusterSet, outlierSet
```

Algorithm 1 shows the pseudo code for the complete DBHD method. The DBHD expects three input parameters: the parameter n defines the minimum size of a cluster. With the ρ parameter, the threshold for this deviation within a cluster is set. The last parameter β defines the maximum deviation of average distances within a cluster.

Figure 4 illustrates the most important steps of the algorithm for the twoDiamond [16] dataset. The dataset consists of two closely spaced diamonds, shown in Figure 4(a). The algorithm has found the current σ of the cluster, which is marked green,

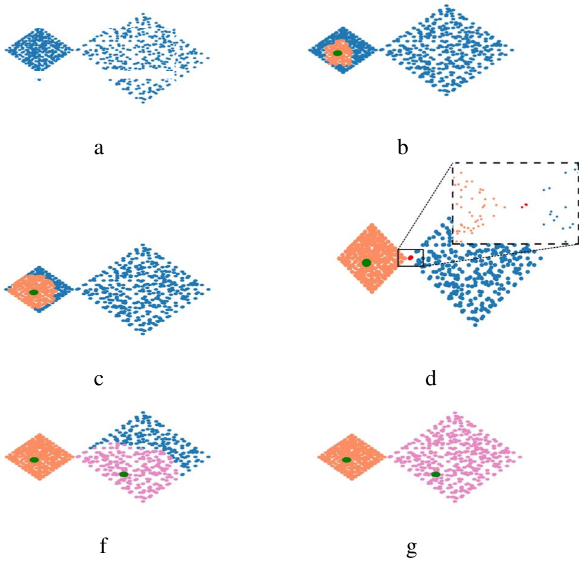

Fig. 4: We illustrate the steps of DBHD on a variation of the twoDiamond dataset [16].

and its ϵ-neighborhood, which builds the potential cluster members P are marked orange (Figure 4(b)). Afterward, further data points are added to the cluster through the loop starting in line 8. All added data points satisfy the conditions according to lines 12 and 13. Figure 4 (a, b, and c) shows how data points belong to the cluster are added gradually. In Figure 4(d), the first cluster is completely extracted. The red data points in this figure are points that are potential members of the cluster (they are in P). However, they do not meet the second criteria: $\rho \times |\mathcal{H}(x, \epsilon) \setminus M| \leq |\mathcal{H}(x, \epsilon) \cap M)|$. Therefore, they are rejected as part of the left cluster. If we would only consider the data points' density at this point—that is the first condition—it would be difficult to decide to which cluster the data points belong. Condition two makes it clear to which cluster the connecting points belong. Since no more data points fulfill both conditions, the left cluster is finished. The algorithm searches for a new σ within the region of the right cluster and data points that belong to it 4(e) and (f). Figure 4(g) shows the final result of the algorithm.

IV. EXPERIMENTS

A. Experimental Setup

In this section, we compare DBHD with other state-of-the-art algorithms using three artificially generated datasets and real-world datasets. The three synthetic datasets shown in Figure 1, Figure 5, and Figure 6, were generated by us. In all three datasets, we have clusters characterized by vastly different shapes, sizes, and densities. Besides the cluster properties, each dataset also contains either no noise (Figure 1), local noise (Figure 5), or global noise (Figure 6).

We normalized all data sets into the range [0,1] and use the Euclidean norm as a distance measure.

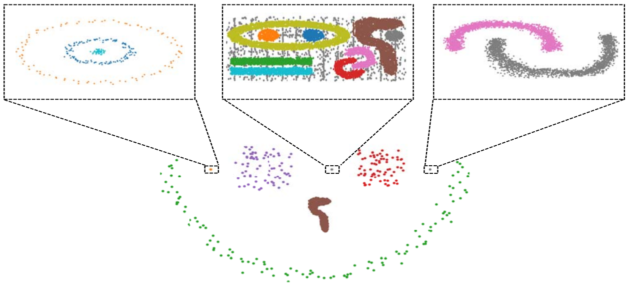

Fig. 5: The synthetic dataset 'ArtificialLocalN' with local noise in a dense region.

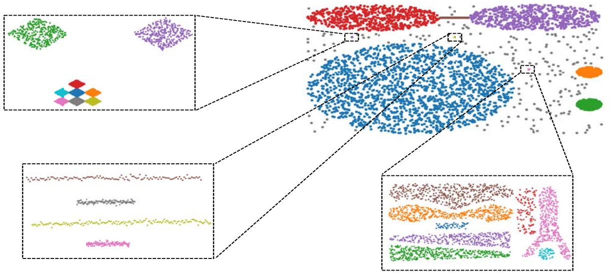

Fig. 6: The synthetic dataset 'ArtificialGlobalN' with global noise.

For algorithms that can mark data points as outliers, we considered these data points as a separate cluster.

B. Performance Evaluation

First, we compare the clustering outcomes of all methods quantitatively. For this, we use two commonly used metrics, the adjusted mutual information (AMI), and the adjusted rand index (ARI) [17]. Table I shows the highest achieved ARI and AMI of each algorithm over a wide range of possible parameters.

1) Artificial dataset: In addition to the vastly different cluster characteristics, two of the artificial datasets also contain local (ArtificialLocalN, shown in Figure 5) and global noise (ArtificialGlobalN, shown in Figure 6). Ideally, clustering algorithms should be robust to both kinds of noise. The results in Table I confirm that DBHD outperforms the state-of-the-art algorithms with an AMI of over 0.93 for both datasets.

2) Real-World Data: As shown in Table I, DBHD outperforms the competing techniques in nine out of fourteen cases. In some scenarios, there is even a difference of more than ten percentage points. In the remaining datasets (besides WISC), it yields the second-highest AMI or ARI value. Although we have covered a large domain of datasets in the experiment, DBHD shows excellent results in almost all areas. This makes our new algorithm an attractive candidate for future data mining tasks.

V. CONCLUSION

We proposed a novel density-based clustering algorithm for determining the density of clusters adaptively. It can handle

TABLE I: Performance Evaluation. The best values are shown in bold and the runner up are underlined

Dataset	Metric	DBHD	DBSCAN	OPTICS	HDBSCAN	*k*-means	SpectralACL	Spectral	DPC	Mean-shift	SNN-DPC	LSDBC	DBADV
ArtificialIntro (Fig. 1)	AMI	**0.990**	0.179	0.708	0.829	0.150	0.152	0.281	0.000†	0.150	0.156	0.084	0.947
	ARI	**0.993**	0.018	0.288	0.430	0.013	0.016	0.047	0.000†	0.028	0.0134	0.094	0.902
ArtificialLocalN	AMI	**0.938**	0.563	0.671	0.652	0.234	0.390	0.545	0.764†	0.556	0.755	0.924	0.563†
	ARI	**0.937**	0.239	0.307	0.279	0.547	0.296	0.235	0.533†	0.238	0.484	0.921	0.239†
ArtificialGlobalN	AMI	**0.987**	0.721	0.768	0.781	0.580	0.600	0.512	0.445†	0.481	0.695	0.423	0.755†
	ARI	**0.983**	0.405	0.403	0.476	0.181	0.340	0.203	0.153†	0.166	0.259	0.776	0.429†
Bank	AMI	**0.920**	0.735	0.313	0.723	0.017	0.817	0.038	0.777	0.301	0.560	0.573	0.876
	ARI	**0.961**	0.826	0.210	0.798	0.022	0.868	0.060	0.815	0.106	0.606	0.579	0.935
Soybean-large	AMI	0.661	0.470	0.561	0.572	0.646	0.418	0.618	0.554	0.516	0.666	0.561	**0.680**
	ARI	0.438	0.129	0.240	0.263	0.419	0.086	0.228	0.343	0.306	0.409	0.225	**0.470**
Dermatology	AMI	0.883	0.602	0.743	0.636	0.877	0.462	0.739	0.390	0.362	**0.927**	0.807	0.800
	ARI	0.841	0.387	0.646	0.456	0.729	0.263	0.555	0.265	0.224	**0.931**	0.769	0.705
Statlog-australian-credit	AMI	**0.433**	0.204	0.219	0.214	0.427	0.104	0.336	0.205	0.204	0.246	0.268	0.215
	ARI	**0.516**	0.119	0.244	0.145	0.504	0.065	0.413	0.101	0.101	0.213	0.327	0.121
Semeion	AMI	**0.663**	0.000	0.383	0.414	0.576	0.000	0.000	0.000	0.000	0.645	0.492	0.176
	ARI	**0.572**	0.000	0.076	0.234	0.480	0.000	0.000	0.000	0.000	0.495	0.315	0.020
Ecoli	AMI	**0.687**	0.577	0.541	0.423	0.584	0.589	0.541	0.664	0.635	0.685	0.602	0.618
	ARI	0.738	0.649	0.601	0.417	0.451	0.516	0.420	0.729	0.655	**0.754**	0.654	0.653
WISC	AMI	0.788	0.789	0.781	0.743	0.742	0.788	0.625	0.651	0.714	**0.806**	0.768	0.605
	ARI	0.872	**0.877**	0.866	0.849	0.844	0.872	0.709	0.776	0.836	**0.877**	0.861	0.702
Pendigits	AMI	**0.812**	0.730	0.657	0.751	0.680	0.763	0.679	0.771†	0.735	0.785	0.800	0.681
	ARI	**0.733**	0.580	0.359	0.592	0.531	0.656	0.484	0.689†	0.651	0.642	0.734	0.514
Multiple-features	AMI	**0.906**	0.000	0.493	0.431	0.780	0.005	0.000	0.000	0.001	0.875	0.684	0.486
	ARI	**0.909**	0.000	0.159	0.677	0.700	0.001	0.000	0.002	0.000	0.804	0.475	0.297
OptDigits	AMI	**0.881**	0.491	0.518	0.685	0.746	0.203	0.760	0.295	0.210	0.863	0.748	0.574
	ARI	**0.869**	0.125	0.154	0.434	0.678	0.015	0.538	0.140	0.032	0.788	0.631	0.454
Letter	AMI	**0.609**	0.457	0.339	0.475	0.360	0.310	0.377	0.502	0.513	0.461	0.499	0.582†
	ARI	**0.252**	0.071	0.009	0.041	0.139	0.094	0.142	0.132	0.116	0.175	0.108	0.189†
USPS	AMI	0.743	0.274	0.293	0.400	0.609	0.211	0.047	0.101†	0.089	**0.769**	0.471	0.525
	ARI	**0.646**	0.063	0.073	0.098	0.534	0.019	0.000	0.245†	0.108	0.618	0.218	0.426
Average	AMI	**0.793**	0.453	0.533	0.582	0.534	0.387	0.407	0.282	0.364	0.660	0.580	0.606
	ARI	**0.750**	0.299	0.309	0.413	0.451	0.274	0.269	0.290	0.238	0.538	0.512	0.470

Values marked with † show the best value obtained within three days.

large datasets with large number of clusters with vastly different properties like extremely different scaling. Furthermore, it provides high parameter stability, and is robust against outliers. We introduced two new conditions to distinguish between different types of data points. On this basis and the adaptively calculated density information, clusters can be filtered more effectively compared to state-of-the-art algorithms. We performed extensive experiments with synthetic and real datasets. The results confirm that our algorithm is effective and outperforms other state-of-the-art algorithms.

References

[1] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[2] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[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. J. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 1, pp. 1–51, 2015.

[5] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *The Journal of Open Source Software*, vol. 2, no. 11, p. 205, 2017.

[6] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[7] R. Liu, H. Wang, and X. Yu, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," *information sciences*, vol. 450, pp. 200–226, 2018.

[8] E. Biçici and D. Yuret, "Locally scaled density based clustering," in *International conference on adaptive and natural computing algorithms*. Springer, 2007, pp. 739–748.

[9] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[10] S. Hess, W. Duivesteijn, P. Honysz, and K. Morik, "The spectacl of nonconvex clustering: a spectral approach to density-based clustering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3788–3795.

[11] L. Qian, C. Plant, and C. Böhm, "Density-based clustering for adaptive density variation," in *IEEE International Conference on Data Mining, ICDM 2021, Auckland, New Zealand, December 7-10, 2021*. IEEE, 2021, pp. 1282–1287. [Online]. Available: https://doi.org/10.1109/ICDM51629.2021.00158

[12] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231–240, 2011.

[13] D. Mautz, W. Ye, C. Plant, and C. Böhm, "Discovering non-redundant k-means clusterings in optimal subspaces," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1973–1982.

[14] ——, "Towards an optimal subspace for k-means," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 365–373.

[15] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[16] T. Barton, "Clustering-Benchmark," https://github.com/deric/clustering-benchmark, 2017, 2020-03-23.

[17] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *The Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.