IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 4, 2018, accepted May 3, 2018, date of publication May 15, 2018, date of current version June 5, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2836389

Density-Based Multiscale Analysis for Clustering in Strong Noise Settings With Varying Densities

TIAN-TIAN ZHANG AND BO YUAN, (Member, IEEE)
Intelligent Computing Lab, Division of Informatics, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China
Corresponding author: Bo Yuan (yuanb@sz.tsinghua.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant U1713214.

ABSTRACT Finding meaningful clustering patterns in data can be very challenging when the clusters are of arbitrary shapes, different sizes, or densities, and especially when the data set contains high percentage (e.g., 80%) of noise. Unfortunately, most existing clustering techniques cannot properly handle this tough situation and often result in dramatically deteriorating performance. In this paper, a purposefully designed clustering algorithm called Density-Based Multiscale Analysis for Clustering (DBMAC)-II is proposed, which is an improved version of the latest strong-noise clustering algorithm DBMAC. DBMAC is proposed under the assumption that all clusters are homogeneous and cannot work well when the data set contains clusters of varying densities. DBMAC-II overcomes the limitation of DBMAC by executing the multiscale analysis iteratively and can conduct strong noise-robust clustering without any strict assumption on the shapes and densities of clusters. In DBMAC-II, each data point or object is mapped into a feature space using its r-neighborhood statistics with different r (radius) values, which is similar to DBMAC. In general, the higher the value of r-neighborhood statistics, the more likely the object is considered as a ''clustered'' object. Instead of trying to find a single optimal r value, a set of radius values appropriate for separating ''clustered'' objects and ''noisy'' objects is identified, using a formal statistical method for multimodality test, referred to as multiscale analysis. For clusters with varying densities, multiscale analysis is applied to extract the clusters with the highest density from the current data set iteratively. Moreover, a statistical uniformity test for measuring clustering tendency is used as the self-adaptive stopping criterion of the iteration. Comprehensive experimental studies on a series of challenging benchmark data sets demonstrate that DBMAC-II is not only superior to classical density-based clustering approaches, including DBSCAN, OPTICS, and HDBSCAN, but also can consistently outperform the latest strong-noise robust clustering techniques, such as Skinny-dip.

INDEX TERMS Multiscale analysis, density-based clustering, heterogeneous clusters, strong noise.

I. INTRODUCTION

Noisy data is prevalent due to various factors such as the intrinsic randomness of the underlying system and errors in measurements. Clustering as an important technique of data analytics for discovering the structure of patterns from unlabeled data has been widely studied in different areas including artificial intelligence, data mining and machine learning [1]. In the real world, it is not unusual that datasets may contain considerably more ''noisy'' objects than ''clustered'' or useful objects and clusters are of arbitrary shapes, different sizes and densities in the same time. Mining valuable information from such datasets is a very challenging task for different parties. Many existing clustering algorithms can produce competitive results when there is no noise or only a small percentage of noise in the data. Nevertheless, in strong noise settings (e.g., 80% noise), it is difficult for them to detect the cluster structure correctly. This is an issue that has received relatively less attention from the research community as most clustering algorithms are not specifically designed with high level noise in mind.

Many traditional clustering methods such as centroid-based approaches (K-means [2], X-means [3]), model-based approaches (EM [4]) and spectral clustering [5] are generally not suitable for handling noise as they produce a partition of the raw input dataset, completely ignoring the existence of noise [6]. For low level noise, DBSCAN [7] employs the concept of density and regards data points from the sparse region as noise, which is significantly more effective in

discovering clusters of arbitrary shapes from data with noise, compared to partitioning-based clustering approaches. However, on datasets with large variation in densities, DBSCAN often do not perform well as a single set of its key parameters (*minPts, Eps*) cannot suit all clusters. The basic idea of OPTICS [8] is similar to DBSCAN but it does not produce clusters explicitly. Instead, it creates an augmented ordering of the dataset representing its density-based clustering structure. After obtaining the points in a particular ordering annotated with their smallest ''reachability distances'', the hierarchical structure of clusters can be obtained using the ''reachability'' plot. HDBSCAN [9] is a hierarchical clustering method, which can be seen as a conceptual and algorithmic improvement over DBSCAN. It can find clusters of varying densities by defining the notion of ''mutual reachability distance'' to represent the relationship of a pair of objects, and proposing an extension of Minimum Spanning Tree (MST) of the Mutual Reachability Graph to construct the cluster hierarchy, and finally extracting the clustering pattern based on the stability of clusters.

SNN [10] redefines the similarity between a pair of points in terms of how many nearest neighbors they share. By using this definition of similarity, SNN alleviates the issue of varying densities and high dimensionality. DECODE [11] is based on a reversible jump Markov Chain Monte Carlo (MCMC) strategy in which a spatial dataset is presumed to consist of different point processes and clusters with different densities belong to different point processes. Recently, a density-ratio based clustering for discovering clusters with varying densities has been proposed [12], which can be implemented in two ways: ReCon is to modify a density-based clustering algorithm to perform density-ratio based clustering by using its density estimator to compute the density-ratio; ReScale involves only the rescaling of the dataset, which is convenient for existing density-based clustering algorithms. However, all of the above algorithms and other noise-robust clustering techniques such as BIRCH [13], CURE [14], SYNC [15], FOSSCLU [16], and cluster tree [17] still suffer from severe performance degradation in strong noise settings.

There has also been some limited progress in clustering methods for datasets with strong noise. Dasgupta and Raftery [18] consider the problem of detecting minefields and seismic faults from ''cluttered'' data. They refine the final partition with the EM algorithm and use approximate Bayes factors to choose the number of clusters, but the technique is restricted to 2D spaces. An alternative implementation [19] of the CFF algorithm [20] is proposed by addressing the computational issue in both the density estimation and the agglomeration steps, which has only been verified on problems with low dimensionality (up to $d=5$). In the area of projected clustering, the divisive projected clustering (DPCLUS) algorithm [21] for detecting correlation clusters in highly noisy data partitions the dataset into clusters in a top-down manner, in order to find a suitable criterion for data partition. Recently, a novel clustering method named Skinny-dip [22] was presented in KDD 2016, which is based on Hartigan's dip test of unimodality [23]. It can effectively handle extremely noisy data with heterogeneous clusters under the assumption that each cluster coincides with the mode of its multivariate distribution. However, it does not work well on clusters with arbitrary shapes as it needs to project the data to each dimension sequentially. To solve this issue, DBMAC [24] augments traditional density-based clustering techniques by introducing multiscale analysis. It is highly competent at tackling clusters of arbitrary shapes in strong noise settings but assumes homogeneous density among clusters.

In some research fields, such as anomaly detection, there are also effective methods for identifying unwanted objects. Anomaly detection is the process of finding data objects featuring significant deviation from the regular pattern of the common data behavior in a specific domain. Generally, it means that these data objects are ''dissimilar'' to other observations in the dataset [25]. A number of unsupervised anomaly detection techniques have been proposed in the literature including density-based techniques (KNN [26], LOF [27], and several variations [28]), subspace-based [29] and correlation-based (COP [30], and LOCI [31]) outlier detection techniques for high-dimensional data, and cluster analysis-based outlier detection techniques (CBLOF [32] and HDBSCAN [9], [33]).

It should be noted that noise removal is related to but distinct from anomaly detection, although both of them deal with unwanted objects in the data. Noise can be defined as a phenomenon in the data, which is not of interest to the analyst but acts as an obstruction to data analysis. A typical example is the large amount of objects spreading uniformly or randomly throughout the data space, just like the type of strong noise clustering problems addressed in this paper. This noisy pattern does not adhere to the common statistical definition of outliers or rare objects, and many outlier detection methods in particular unsupervised techniques will fail on such datasets.

In this paper, we further extend the applicability of DBMAC and propose DBMAC-II to discover clusters of varying densities from highly noisy datasets. The core idea is to apply multiscale analysis iteratively to sequentially extract potential clusters of different densities. After extracting ''clustered'' objects with the same level of density in each iteration, we execute standard density-based clustering such as DBSCAN to obtain the partial results. A formal statistical test is used as the self-adaptive stopping criterion of the overall process. Section II introduces four clustering algorithms including DBSCAN, OPTICS, HDBSCAN and Skinny-dip with preliminary experiment results to reveal their limitations. In Section III, we review the key elements of DBMAC for homogeneous clusters and present the details of the new DBMAC-II for clusters with varying densities in highly noisy datasets. Section IV contains systematic experimental studies on the effectiveness of DBMAC-II in various settings. This paper is concluded in Section V with some analysis and directions for future work.

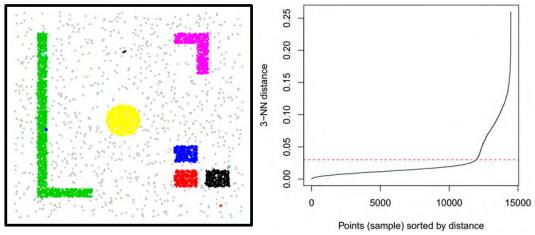

FIGURE 1. Clustering result of DBSCAN with 20% noise and homogeneous clusters, *Eps*=0.030, *minPts*=3, AMI=0.975.

II. RELATED WORK

A. DBSCAN

DBSCAN [7] is one of the most popular clustering algorithms, which can find clusters of arbitrary shapes and has a clear definition of noise. In DBSCAN, the density of a point is obtained by counting the number of points within a specific distance (radius), *Eps*, from its location. Points with densities above the threshold, *MinPts*, are classified as core points, while noisy points are defined as non-core points that do not have a core point within the specific radius.

An effective heuristic for determining the parameter values of *Eps* and *MinPts* is *sorted k-dist graph*. The idea is to calculate the average distance of each point to its k nearest neighbors, where the value of k (corresponding to *MinPts*) is specified by user, which is usually set to data dimensionality plus one. In *sorted k-dist graph*, the position where a sharp change occurs is chosen as the optimal *Eps* value. This estimation is generally reasonable when the amount of noise is small and all clusters have the same density but can be problematic when the percentage of noise is high and clusters are of different densities. In Figure 1, when the dataset contains 20% noise and all clusters have the same density, the estimated *Eps* from *sorted k-dist graph* can result in a good separation of clusters and noise. By contrast, when the percentage of noise increases to 70% and the densities of clusters are different, it becomes very tricky and almost impossible to choose a single combination of parameters appropriately for all clusters and noise. After fine-tuning its parameters,[1] as shown in Figure 2, DBSCAN produced 367 clusters in this case (AMI[2] 0.460), primarily because there were many areas where the density threshold was exceeded, due to the randomness of heavy noise. Meanwhile, it is intuitive to use a large *Eps* to prevent the fragmentation of the thin rectangular cluster in the left, but it may also cause the three neighboring clusters in the right-bottom corner to merge and more noise to be recognized as clusters. Consequently, it is most likely that no parameter setting of DBSCAN can properly solve such complicated situations.

B. OPTICS

OPTICS [8] is another well-known density based clustering algorithm, which overcomes the weakness of DBSCAN on handling clusters of varying densities. It employs a unique approach by drawing a ''reachability'' plot that reveals the clusters in different regions with respect to the local densities. The plot is produced by a linear order of all objects where spatially adjacent objects are close to each other such that object x_i is the closest to x_{i-1} in terms of the ''reachability distance'', and the first object x_0 is chosen randomly. In addition, it records the ''reachability distance'' of each object. In the ''reachability'' plot, all objects are distributed according to the linear order along the x-axis, and the y-axis indicates the ''reachability distance''. As a cluster center normally has higher density or lower ''reachability distance'' than the cluster boundary, each cluster is visually represented as a ''valley'' in this plot, and the clusters can be extracted by a hierarchical method with the optimal steepness threshold ξ of the reachability plot. OPTICS can create the reachability plot for any datasets with varying-densities clusters. Meanwhile, this reachability plot is rather insensitive to the input parameters of the method (the generating distance *Eps* and the value for *minPts*).

As shown in Figure 3(a), when the dataset has low level of noise, OPTICS can successfully recognize most noise and clusters of varying densities. Nevertheless, a ''valley'' in the reachability plot may contain instances from different clusters if some clusters are very close to each other or their boundaries overlap with each other, leading to a faulty result. More seriously, with the increasing level of noise, the ''valleys'' gradually become more unobservable because the density of noise is also very high, and the ''reachability distance'' of the cluster boundary is not distinctly larger than that of the cluster center. As a result, the performance of OPTICS on datasets with high-level noise may be unsatisfactory, as shown in Figure 2, which presents the best result (AMI 0.576) by fine-tuning its parameters.[3]

C. HDBSCAN

HDBSCAN [9] is a hierarchical density-based clustering algorithm, which is a variation of DBSCAN designed to overcome one of the hardest issues in clustering: the detection of clusters with varying densities. This hierarchical version performs DBSCAN with varying values of radius *Eps* and integrates all results to find the best clustering solution. In the process of clustering, HDBSCAN firstly transforms the space according to the density by defining the notion of ''mutual reachability distance'' to represent the distance between a pair of objects. Then, it finds the MST of all points, where for each pair of points there is an edge connecting them with mutual reachability distance as the weight. Finally, it performs single

[1] Here $Eps = 0.035$ and $minPts = 3$.

[2] AMI (Adjusted Mutual Information), a variation of mutual information, is used as clustering quality measurement: For a dataset $\mathcal{X}$ of n elements, $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, given the true set of clusters $U = \{U_1, U_2, \ldots, U_R\}$ and the clusters obtained from some clustering algorithm $V = \{V_1, V_2, \ldots, V_C\}$, AMI is defined as follows: $AMI(U, V) = \frac{MI(U,V) - E\{MI(U,V)\}}{\max\{H(U), H(V)\} - E\{MI(U,V)\}}$, where $MI(U, V)$ is the mutual information (MI) between two partitions, $EMI(U, V)$ is the expected mutual information between two random clusterings, $H(U)$ and $H(V)$ are entropies.

[3] Here $Eps = 1$, $minPts = 15$, and $\xi = 0.030$.

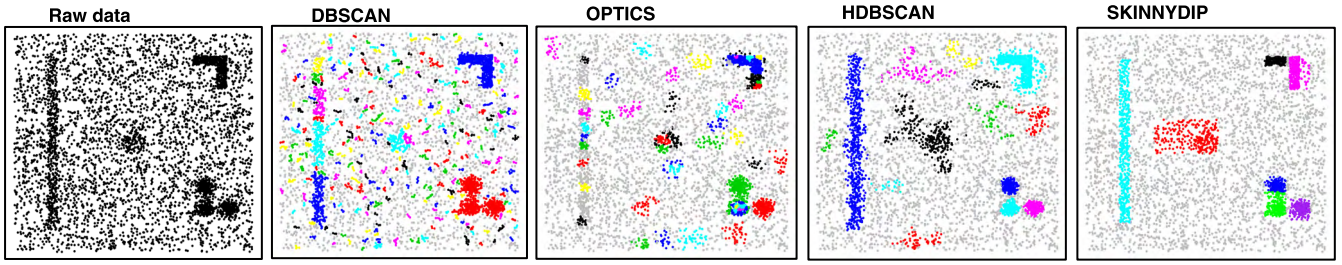

FIGURE 2. Clustering results of various techniques on a dataset with 70% noise and heterogeneous clusters.

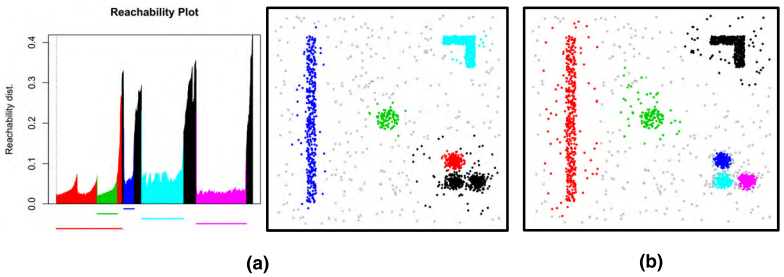

FIGURE 3. Clustering results of OPTICS and HDBSCAN on a dataset with 20% noise and heterogeneous clusters. (a) OPTICS: *Eps*=1, *minPts*=15, ξ=0.090, AMI=0.788. (b) HDBSCAN: AMI=0.857.

linkage clustering on the transformed space. Instead of taking a single *Eps* value as the cut-off level for the dendrogram, HDBSCAN cuts the tree at different heights to select varying-densities clusters based on cluster stability.

As shown in Figure 3(b), when the dataset has low level of noise, HDBSCAN can correctly detect most noise and clusters of varying densities except for some noisy objects neighboring to clusters. With the increasing level of noise, the mutual reachability distance of noise objects in MST will decrease gradually and even become close to clustered objects in many areas due to the randomness of high level of noise. In this case, if HDBSCAN splits these noisy objects in the dense area, the value of the cluster's stability will become smaller. As a result, HDBSCAN tends to regard them as many small clusters. The result in Figure 2 shows that HDBSCAN (AMI 0.701) performs better than DBSCAN and OPTICS in dataset with high level of noise, but there are also several noisy objects incorrectly identified as clusters.

D. SKINNYDIP

Skinny-dip [22] is explicitly proposed as a clustering algorithm for datasets with strong noise, which is claimed to be highly noise-robust, practically parameter-free and completely deterministic. Based on Hartigan's elegant dip test of unimodality, the authors proposed a recursive heuristic to extract clusters in noisy datasets, assuming that each cluster admits a unimodal shape.

In Figure 4, for the horizontally projected univariate data, dip test is initially executed on all samples $x_1 \leq x_2 \leq \ldots \leq x_n$, and the identified modal interval spans

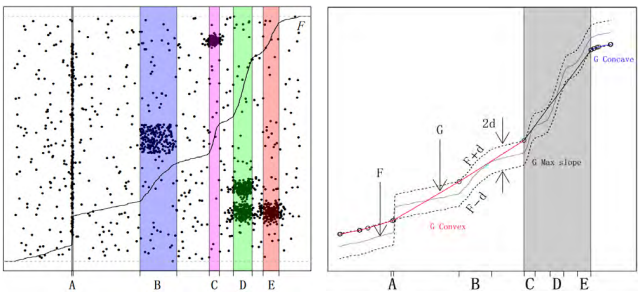

FIGURE 4. Dip solution (right) for the horizontal-axis projection of the dataset (left, 30% noise).

three modes C, D, E (the gray region). Then, according to the location of the modal interval $[x_L, x_U]$, the algorithm works within this interval recursively to extract inner individual modal intervals $[x_{L_C}, x_{U_C}]$, $[x_{L_D}, x_{U_D}]$, and $[x_{L_E}, x_{U_E}]$. Next, the search range turns to the left and right sides of $[x_L, x_U]$ respectively, until all modes are found. This process (UNIDIP) is the core part of Skinny-dip, suitable for univariate clustering. Based on UNIDIP, a recursive procedure over the dimensions of the data space is used to generalize to multivariate cases. Due to the property of projected clustering and the hypothesis testing of unimodality, Skinny-dip can only detect clusters that take a unimodal form along each coordinate and produce clusters of regular shapes. When clusters are of irregular shapes or their projections overlap, the limitation of this type of technique becomes evident. In Figure 2, Skinny-dip can only output the rectangle-shaped clusters in 2D spaces, regardless of the real shapes of clusters (AMI 0.755).

III. DENSITY-BASED MULTISCALE ANALYSIS FOR CLUSTERING

The motivation of our work is to develop a competent clustering technique that can reliably identify clustering patterns of arbitrary shapes and different densities in extremely noisy settings. The main idea is to design a robust density-based criterion to filter out the noise and then apply existing clustering algorithms such as DBSCAN to do the final clustering for extracted "clustered" objects.

A. PROBLEM FORMULATION

Formally, let $\mathcal{X}$ be a dataset of n data objects (row vectors) with d-dimensional features (column vectors) and the percentage of noise that $\mathcal{X}$ contains is significant (e.g., 80%). $\boldsymbol{x}_i \in \mathcal{X}$ stands for the i^{th} object in $\mathcal{X}$ where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})$. To simplify the problem, we assume that the set of noisy objects coincides with a uniform distribution. For this noisy data, the functionality of the clustering algorithm is as follows:

- **Input:** Data $\mathcal{X} \in R^{n\times d}$, $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$.
- **Output:** $\{C_1, C_2, \ldots, C_k, \Phi\}$ $(k \geq 1)$, a partition of $\mathcal{X}$ where Φ represents the set of noise and $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ is the set of k normal clusters.

Referring to Figure 5, we give some definitions as follows:

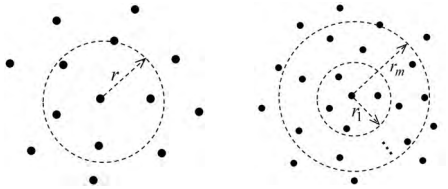

FIGURE 5. An example of single-neighborhood statistics (left) vs. multiscale-neighborhood statistics (right).

Definition 1 (Single-Neighborhood Statistics): The *r-neighborhood* of a point $\boldsymbol{x}_i$, denoted by $s_r(\boldsymbol{x}_i)$, is defined as:

$$s_r(\boldsymbol{x}_i) = \{\boldsymbol{x}_j \in \mathcal{X} | dist(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq r, i \neq j\} \quad (1)$$

Based on (1), the *r-neighborhood statistics* of point $\boldsymbol{x}_i$ is denoted as:

$$n_r(\boldsymbol{x}_i) = card(s_r(\boldsymbol{x}_i)) \quad (2)$$

Definition 2 (Multiscale-Neighborhood Statistics): The *multiscale-neighborhood* of a point $\boldsymbol{x}_i$, denoted by $S_R(\boldsymbol{x}_i)$, is defined as:

$$S_R(\boldsymbol{x}_i) = \{s_{r_1}(\boldsymbol{x}_i), s_{r_2}(\boldsymbol{x}_i), \ldots, s_{r_m}(\boldsymbol{x}_i)\} \quad (3)$$

where $R = \{r_1, r_2, \ldots, r_m\}$ is the set of radius values that forms an arithmetic sequence:

$$R = \{minr, minr + \Delta, minr + 2\Delta, \ldots, maxr\} \quad (4)$$

For each $\boldsymbol{x}_i$, its *multiscale-neighborhood statistics* is then denoted as:

$$N_R(\boldsymbol{x}_i) = \{n_{r_1}(\boldsymbol{x}_i), n_{r_2}(\boldsymbol{x}_i), \ldots, n_{r_m}(\boldsymbol{x}_i)\} \quad (5)$$

B. DBMAC: SINGLE-DENSITY CLUSTERING

In strong noise settings, the difference in density between noise and clusters can be much smaller compared to low-level noise situations, which makes the discriminative information of clusters and noise less obvious. In this case, although there may exist a single r value that can be used to get satisfactory partition results for a dataset with completely homogeneous clusters, in practice, it is likely to be very challenging to obtain such an optimal r. Furthermore, the densities of clusters and noise are often not completely homogeneous for real-world problems and a fixed r value is not expected to work well in such situations. In order to obtain more indicative features of clusters and noise, we proposed to use multiscale analysis to map the original data points into a feature space with richer information to divide clusters and noise, instead of using a single r value [23]. This is the core mechanism of DBMAC, which can produce high quality results compared to existing techniques. However, it only considers the situation of homogeneous clusters.

To analyze the effect of different r values, we take the data in Figure 6 as an example. For each $\boldsymbol{x}_i$ in $\mathcal{X}$, we calculated its *r-neighborhood statistics* $n_r(\boldsymbol{x}_i)$ with a set of radius values $\{r_1, r_2, \ldots, r_m\}$. Figure 7 shows the distribution of *r-neighborhood statistics* with a series of increasing r values for the example data. We can see that there is only one major mode in the density curve when r is very small (the *r-neighborhood statistics* features are not discriminative). Then, as the increase of r value, the density distributions do present two distinct modes (two levels of densities) within a certain range of r values. In fact, the bimodal patterns approximately correspond to the single-density clusters and noise, respectively. However, as r keeps increasing, the bimodal curve will become deformed.

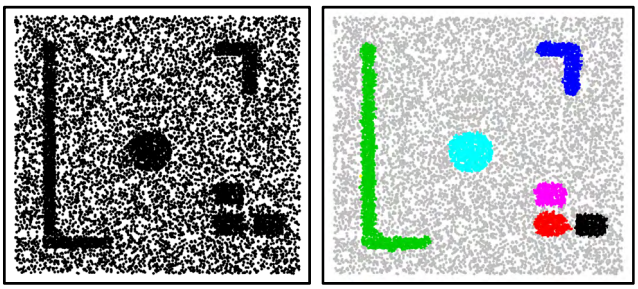

FIGURE 6. An example dataset (left) with single-density clusters and the result of DBMAC (right, *Eps* = 0.030, *minPts* = 3, AMI=0.904).

The procedure of multiscale analysis is described in Algorithm 1. The *r-neighborhood statistics* $\overrightarrow{N_r}$ with a small initial r value *minr* for each $\boldsymbol{x}_i$ in $\mathcal{X}$ is calculated first (lines 3 to 5). Then, the unimodality test using Hartigan's dip test is conducted and the number of modes is obtained using UNIDIP (see Section 2.3) to identify the discriminatory feature matrix suitable for partitioning "clustered" objects and "noisy" objects (lines 6 to 13). If the distribution of $\overrightarrow{N_r}$ is unimodal, which means that the current

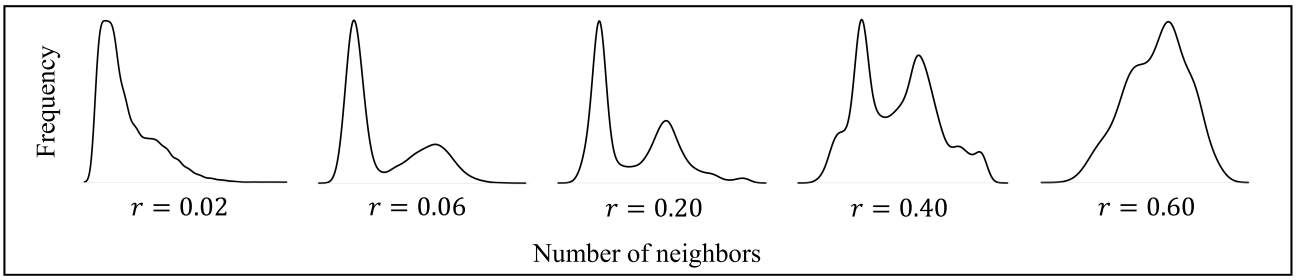

FIGURE 7. The distribution of *r-neighborhood statistics* with a series of increasing *r* values for the example data.

TABLE 1. Partitioning results of "clustered" and "noisy" objects with different r values (C: 4731; N: 8136).

r value	0.02	*0.04	*0.08	*0.12	*0.16	*0.20	0.30	0.40	0.50	0.60	0.70	0.80
TP	3310	4293	4538	4664	4714	4710	4652	4627	4526	4449	4473	4271
FP	124	44	105	266	512	738	1372	1909	2302	2988	3659	4086
TN	8012	8092	8031	7870	7624	7398	6764	6227	5834	5148	4477	4050
FN	1421	438	193	67	17	21	79	104	205	282	258	460
G-mean	0.830	0.950	0.973	0.977	0.966	0.951	0.904	0.865	0.828	0.771	0.721	0.670

r-neighborhood statistics feature is not discriminative, we discard it and proceed to the next r value. Otherwise, we identify the number $\mathcal{N}$ of modes contained in the current density curve using the UNIDIP algorithm and add the corresponding *r-neighborhood statistics* into the final feature matrix $\overrightarrow{N_R}$. In addition, we assign the $\mathcal{N}$ value to N, representing the number of modes contained in the final feature matrix.[4]

Repeat the above procedures with the next r value with Δ step size until the distribution of *r-neighborhood statistics* becomes deformed and the number of modes is not equal to N. Assuming there are m features corresponding to m consecutive r values containing N modes, the final discriminatory feature matrix obtained is as follows:

$$\overrightarrow{N_R} = \begin{bmatrix} n_{r_{i+1}}(\boldsymbol{x}_1) & \cdots & n_{r_{i+m}}(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ n_{r_{i+1}}(\boldsymbol{x}_n) & \cdots & n_{r_{i+m}}(\boldsymbol{x}_n) \end{bmatrix} \tag{6}$$

In DBMAC (Algorithm 2), k-means is applied with k=2 on the feature space created by multiscale analysis. The cluster with larger cluster center corresponds to all single-density clusters, and the other one corresponds to noise. Figure 8 shows the effectiveness of noise removal for the example dataset in Figure 6. The r range identified by multiscale analysis was [0.04, 0.22], and the *G-mean* value of the results in Figure 6 was 0.978. Table 1 shows the partitioning results[5] of "noisy" and "clustered" objects for the example dataset ("clustered" objects: 4731, "noisy" objects: 8136) with different r values from 0.02 to 0.80. Experiment results show that only those *r-neighborhood statistics* features corresponding to r values (with *) that produced the discriminative bimodal patterns can separate the "clustered" objects and "noisy" objects effectively. Note that DBMAC was very close to the partitioning result produced by the best single r value (0.12) listed in Table 1.

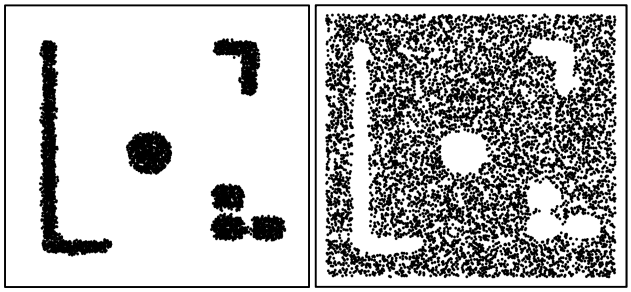

FIGURE 8. The result of multiscale analysis: the single-density clusters component (left) and noise component (right).

After noise removal, the raw dataset is transformed into a new subset containing vast majority of the original "clustered" objects with possibly only a small amount of noise. Next, any clustering algorithm that is weak noise robust and can detect arbitrarily shaped clusters can be used to discover the true clusters. Figure 6 (right) is the clustering result of DBMAC using DBSCAN in the final clustering.

C. DBMAC-II: THE VARYING-DENSITIES CASE

In the previous section, we reviewed the framework of DBMAC, which assumes that all clusters are of the same density. However, this is often unrealistic for real-world problems. Figure 9 is the results of DBMAC on the varying-densities datasets in Figure 2 and Figure 13, which indicates

[4] N refers to the number of modes contained in the first non-unimodal density curve. In single-density clusters situation, there are two levels of density in data, corresponding to noise and all clusters. In this case, $N = 2$.

[5] *G-mean* is an evaluation index of unbalanced learning: $G - mean = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}}$.

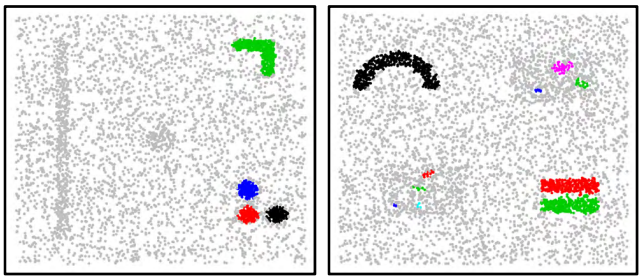

FIGURE 9. The results of DBMAC on two heterogeneous datasets (Figure 2 & Figure 13).

that DBMAC can only discover the higher-density clusters while the lower-density clusters are regarded as noisy objects. In this section, we present DBMAC-II to solve this problem and the objective is to discover clusters of different densities from strongly noisy data.

By applying multiscale analysis to the datasets that contain clusters of two different densities (Figure 2 & Figure 13), we can observe that, as the increase of r, there exist two different patterns of the distribution of *r-neighborhood statistics*, as shown in Figure 10. In the first case, the density curve becomes bimodal from the unimodal pattern. The mode with larger horizontal axis value represents clusters with higher density and the other one corresponds to the combination of clusters with lower density and noise, shown in Figure 10(a). This happens because the density difference between clusters with lower density and noise is quite small. In this case, if we use the k-means algorithm ($k = N = 2$) to make a partition, we can only retrieve the clusters with higher density. In the second case, the density curve becomes trimodal from unimodal distribution when the density difference between clusters with the lower density and noise is large. These three modes approximately correspond to the set of "noisy" objects, the lower-density clusters, and the higher-density clusters respectively, shown in Figure 10(b). In this case, when using k-means ($k = N = 3$) to make the partitioning, we can see that the higher-density clusters are identified correctly, but the middle mode not only contains the lower-density clusters, but also contains some "noisy" objects that are neighboring to the higher-density clusters.

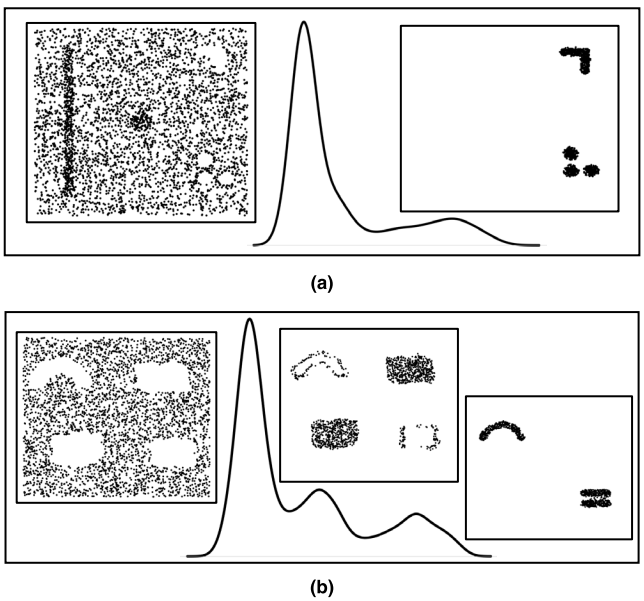

FIGURE 10. Two different cases of the distribution of *r-neighborhood statistics* for the datasets in Figure 2 (case 1) and Figure 13 (case 2). (a) Case 1: small density difference. (b) Case 2: large density difference.

Algorithm 1 Multiscale Analysis (MA)

Input : Data $\mathcal{X} \in \mathbb{R}^{n\times d}$, $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, r_{min}, r_{max}, step size Δ, significance level α.
Output: $\overrightarrow{N_R}$: feature matrix, N: the number of modes contained in $\overrightarrow{N_R}$.

1: $r \leftarrow r_{min}$, $\overrightarrow{N_R} \leftarrow \emptyset$;
2: **while** $r < r_{max}$ **do**
   /* r-neighborhood statistics */
3: **for** $\boldsymbol{x}_i \in \mathcal{X}$ **do**
4: $\overrightarrow{N_r}$.push(card $\left(\left\{\boldsymbol{x}_j \in \mathcal{X} \mid \text{dist}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \le r\right\}\right)$);
5: **end**
   /* create the discriminatory feature matrix */
6: $\overrightarrow{N_r}' \leftarrow$ sort$(\overrightarrow{N_r})$; /* ascending sort */
7: $p \leftarrow$ dip.test$(\overrightarrow{N_r}')$;
8: **if** $p < \alpha$ **then do**
9: $\mathcal{M} \leftarrow$ UNIDIP$(\overrightarrow{N_r}')$, $\mathcal{N} \leftarrow$ length(M);
10: **if** $\overrightarrow{N_R} == \emptyset$ **then do** $\overrightarrow{N_R}$.push$(\overrightarrow{N_r})$, $N \leftarrow \mathcal{N}$;
11: **else** $\mathcal{N} == N ? \overrightarrow{N_R}.\text{push}\left(\overrightarrow{N_r}\right)$: **return** $\{\overrightarrow{N_R}, N\}$;
12: **else if** $\overrightarrow{N_R}! = \emptyset$ **then return** $\{\overrightarrow{N_R}, N\}$;
13: $r \leftarrow r + \Delta$;
14: **end**
15: **return** $\{\overrightarrow{N_R}, N\}$;

From the above analysis, we can see that the highest-density clusters in the current dataset can be identified directly using multiscale analysis. Thus, for varying-densities cases, we propose to execute multiscale analysis iteratively to extract clusters from strongly noisy data, which is the key idea of DBMAC-II (Algorithm 3). In each iteration, we conduct multiscale analysis on the current dataset to identify discriminative features that present the same number of modes firstly (line 4). Then, from the clustering results of k-means ($k = N$), we extract the cluster with the largest cluster center value, corresponding to clusters with the highest density in the current data (line 5 to line 6). For this extracted set of "clustered" objects with the same density, DBMAC-II also uses DBSCAN for final clustering (line 8). As for the remaining data without the highest-density clusters, clusters with the second highest density become the clusters with the highest density in the remaining dataset.

Before the next iteration, in order to keep the noise uniform, the operation of gap filling is executed. In DBMAC-II,

Algorithm 2 DBMAC

Input : Data $\mathcal{X} \in \mathbb{R}^{n\times d}$, $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, r_{min}, r_{max}, step size Δ, significance level α.
Output: Clusters set $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$.

1: $\mathcal{X} \leftarrow$ normalize($\mathcal{X}$);
/* multiscale analysis to identify feature matrix */
2: $\{\overrightarrow{N_R}, N\} \leftarrow \mathrm{MA}(\mathcal{X}, r_{min}, r_{max}, \Delta, \alpha)$;
/* separate clusters from noise */
3: $\{\Omega_1, \Omega_2\} \leftarrow$ k-means $(\overrightarrow{N_R}, 2)$;
4: $cObj \leftarrow \underset{\Omega\in\{\Omega_1,\Omega_2\}}{\text{argmax}}$ (Ω_center);
5: $nObj \leftarrow \underset{\Omega\in\{\Omega_1,\Omega_2\}}{\text{argmin}}$ (Ω_center);
6: $\{C_1, C_2, \ldots, C_k\} \leftarrow$ dbscan($cObj$, eps, $minPts$);
7: **return** $\{C_1, C_2, \ldots, C_k\}$;

gap filling is implemented by random sampling from the clusters extracted from the last iteration, based on the density ratio of noise and extracted clusters (line 10 and line 11). The results of gap filling in the first iteration for the two examples (Figure 2 & Figure 13) are shown in the left side of Figure 11. The resampled data is then used in the next iteration (shown in the right side of Figure 11). Repeat the above procedures until the remaining dataset after gap filling is relatively uniform.

Algorithm 3 DBMAC-II

Input : Data $\mathcal{X} \in \mathbb{R}^{n\times d}$, $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, r_{min}, r_{max}, step size Δ, significance level α.
Output: Cluster set $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$.

1: $\mathcal{X} \leftarrow$ normalize($\mathcal{X}$), $\mathcal{C} \leftarrow \emptyset$;
/* testing the uniformity of data */
2: $p \leftarrow$ MST.test($\mathcal{X}$);
3: **while** $p < \alpha$ **do**
4: /* multiscale analysis to identify feature matrix */
5: $\{\overrightarrow{N_R}, N\} \leftarrow \mathrm{MA}(\mathcal{X}, r_{min}, r_{max}, \Delta, \alpha)$;
6: $[\Omega_1, \Omega_2, \ldots, \Omega_N] \leftarrow$ k-means$(\overrightarrow{N_R}, N)$;
7: $cObj \leftarrow \underset{\Omega\in\{\Omega_1,\Omega_2,\ldots,\Omega_N\}}{\text{argmax}}$ (Ω_center);
8: $nObj \leftarrow \underset{\Omega\in\{\Omega_1,\Omega_2,\ldots,\Omega_N\}}{\text{argmin}}$ (Ω_center);
9: $\{C_1, C_2, \ldots, C_{ki}\} \leftarrow$ dbscan($cObj$, eps, $minPts$);
10: $\mathcal{C} \leftarrow \mathcal{C} \cup \{C_1, C_2, \ldots, C_{ki}\}$;
11: $den\_Rat \leftarrow cObj\_center/nObj\_center$;
12: $\mathcal{X} \leftarrow \mathrm{C}_{\{\Omega_1,\Omega_2,\ldots,\Omega_N\}}(cObj) \cup$ sample($cObj$,den_Rat);
/* uniformity test for new data $\mathcal{X}$ */
13: $p \leftarrow$ MST.test($\mathcal{X}$);
14: **end**
15: **return** $\mathcal{C}$;

In order to determine whether the remaining dataset after gap filling follows the uniform distribution, in DBMAC-II, we employ Friedman-Rafsky's minimal spanning tree (MST) based test [35] for uniformity testing of multi-dimensional data, which is a hypothesis test for measuring the clustering

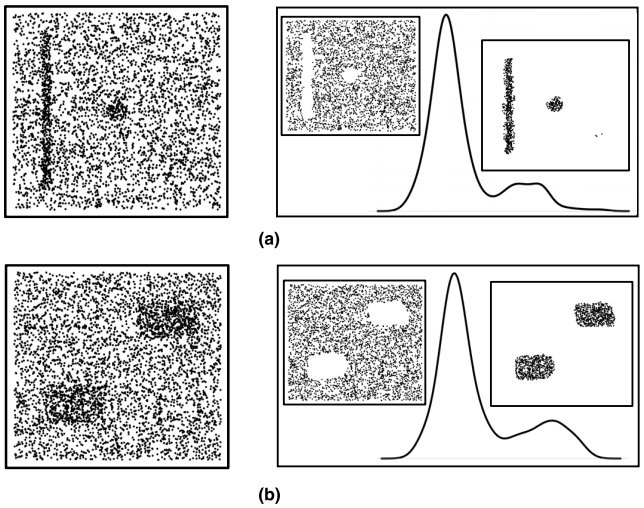

FIGURE 11. The remaining dataset after gap filling (left) and the results of multiscale analysis in the next iteration (right). (a) Case 1 (b) Case 2.

tendency. Firstly, a uniformly distributed sample is generated over a set that approximates the convex hull of the data under testing. Then, the test determines whether this generated sample and the given data belong to the same population. If the null hypothesis that the two samples belong to the same population is accepted, we can say that the dataset of interest is uniformly distributed over the convex hull.

In the context of our situation, the points labeled X are the M_1 given data points after gap filling, and the M_2 points labeled Y are uniformly generated, which approximate the convex hull of the given data. Then, the test for uniformity against a clustered alternative is conducted as follows. Reject the data as uniform when:

$$\frac{T - E[T]}{\sqrt{var[T|C]}} < Z(\mathfrak{L}) \tag{7}$$

where $Z(\mathfrak{L})$ is the $\mathfrak{L}$ quantile of the standard normal distribution. T is the X-Y join count in MST with

$$\begin{aligned} E[T] &= \frac{2M_1M_2}{L} \\ var[T|C] &= \frac{2M_1M_2}{L(L-1)}\{\frac{2M_1M_2 - L}{L} + \frac{C - L + 2}{(L-2)(L-3)} \\ &\quad \times [L(L-1) - 4M_1M_2 + 2]\} \end{aligned} \tag{8}$$

where C is the number of edge pairs in the MST sharing a common node and $L = M_1 + M_2$.

If the remaining data after gap filling is determined as a uniform distribution, it means that there are no clusters hidden in the remaining dataset. In this situation, we stop the iteration and make a combination of all clusters recognized from all previous iterations. The result of DBMAC-II on the dataset in Figure 2 is shown in Figure 12 while Figure 13 is the result of DBMAC-II on another heterogeneous dataset. In both cases, DBMAC-II successfully found all clusters while filtering out the random noise reliably.

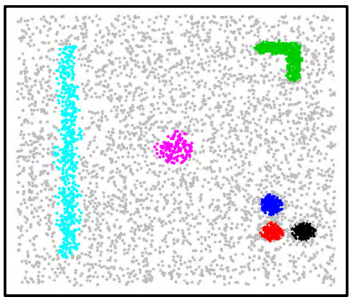

FIGURE 12. The result of DBMAC-II on the heterogeneous dataset shown in Figure 2 ($Eps_1 = 0.035$, $Eps_2 = 0.060$, $minPts = 3$, AMI=0.862).

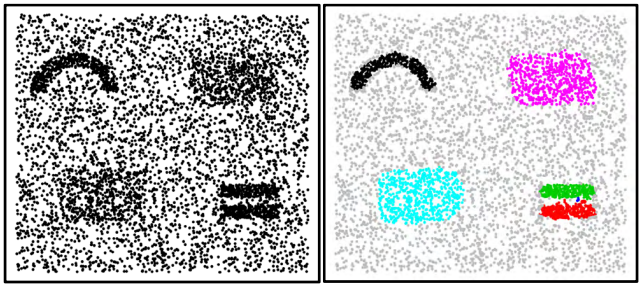

FIGURE 13. Another heterogeneous dataset (left) and the result of DBMAC-II (right, $Eps_1 = 0.035$, $Eps_2 = 0.060$, $minPts = 3$, AMI=0.854).

IV. EXPERIMENTAL EVALUATION

In this section, we conducted comprehensive experimental studies of DBMAC-II, compared to state-of-the-art clustering techniques including density-based algorithms DBSCAN, OPTICS and HDBSCAN, as well as the strong noise-robust algorithm Skinny-dip. DBMAC-II was implemented in R and the standard DBSCAN, OPTICS, and HDBSCAN codes in R were used while the source code of Skinny-dip in R was retrieved from *GitHub* uploaded by its author.

A. 2D SYNTHETIC DATASETS

All 2D synthetic datasets were customized based on selected datasets from *GitHub clustering benchmarks*. These datasets consist of different numbers of irregular clusters and have been used frequently in clustering research. During preprocessing, we modified the densities of clusters and added extra random noise to the level between 70% and 80%. All datasets contained clusters with two significantly different levels of densities, as shown in the first row of Figure 14. The following five rows present the clustering results of DBSCAN, OPTICS, HDBSCAN, Skinny-dip and DBMAC-II, respectively.

DBMAC-II used the default value of $\alpha = 0.05$ in the dip test for unimodality and the MST-based test for uniformity in all cases. To make the comparison as fair as possible, we systematically varied the parameters of other algorithms and presented the best AMI results. For DBSCAN, we fixed $minPt = 3$ (the recommended value: dimensionality plus one) and ran DBSCAN with $Eps = \{0.005, 0.010, \ldots, 0.500\}$. For OPTICS, we used the same parameters (Eps, $minPts$) as DBSCAN with steepness threshold $\xi = \{0.001, 0.002, \ldots, 0.100\}$ to identify clusters hierarchically.

Our results on the four 2D synthetic datasets showed that DBMAC-II outperformed its competitors by a large margin. According to Figure 14, DBSCAN detected almost all "clustered" objects in these highly noisy datasets, including arbitrarily shaped clusters. However, it also made many mistakes by assigning "noisy" objects to many small clusters and fragmenting the true clusters with lower densities, leading to very messy results. The results of OPTICS were also disordered in strong noise settings. Due to the small density difference between clusters and noise, the reachability distance of the cluster boundary was not distinctly larger than that of its cluster center, which makes it hard to find an optimal steepness of the reachability plot to extract clusters hierarchically. In order to get rid of noisy objects, it is intuitive to use a small ξ to separate clusters and noise, but it may also cause the fragmentation of true clusters. Meanwhile, the local density threshold was exceeded in several areas due to the randomness of noise and many "noisy" objects were identified as small clusters.

The results of HDBSCAN were much better than DBSCAN and OPTICS on these four datasets. It correctly detected most noise and clusters of varying densities except for some noisy objects neighboring to clusters and in dense areas. Skinny-dip also filtered out most of the noise, but it cannot identify irregular clusters due to the limitation of projection-based method. In fact, the cluster regions produced by Skinny-dip were all horizontal or vertical to the projected directions, which were significantly different from the true clusters. Meanwhile, some clusters located in the area of relatively low density along the projected direction were mistaken for noise. By contrast, apart from a few flaws along the edges of clusters, DBMAC-II not only filtered out most of "noisy" objects but also correctly identified arbitrarily shaped and clustering of varying densities.

B. MULTI-DIMENSIONAL SYNTHETIC DATASETS

For multi-dimensional cases, we created synthetic datasets using Gaussian distributions and varied the number of points in each cluster to achieve different densities. By default, we generated 11 hyper-sphere clusters with two different densities in three dimensions, in which there were 4 clusters of 600 objects each with a standard deviation of 0.015 and 7 clusters of 500 objects each with a standard deviation of 0.030. The centers of the Gaussians were generated randomly within $[0.1, 0.9]^d$ and the distance between any two centers was greater than 6σ to minimize the chance of overlapping. Noise was randomly generated within $[0, 1]^d$, and 80% of the objects in each dataset were "noisy" objects by default. With the above parameter settings, we hence obtained the default dataset with $n = 29,500$. Dataset parameters varied in the experiments included dimensionality d, the number of clusters k, the noise percentage η and the number of density levels l.

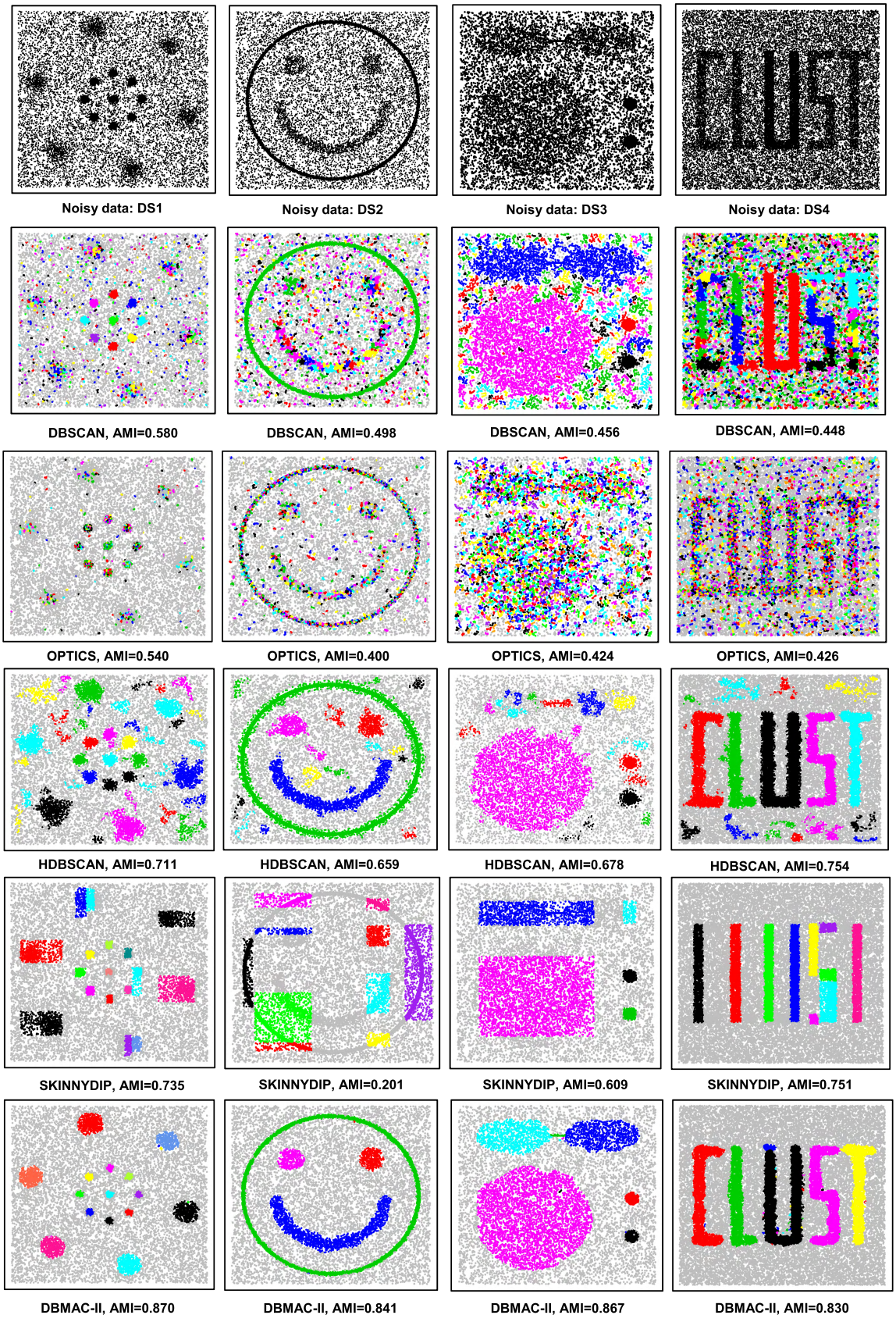

FIGURE 14. Clustering results of five algorithms on four 2D synthetic datasets.

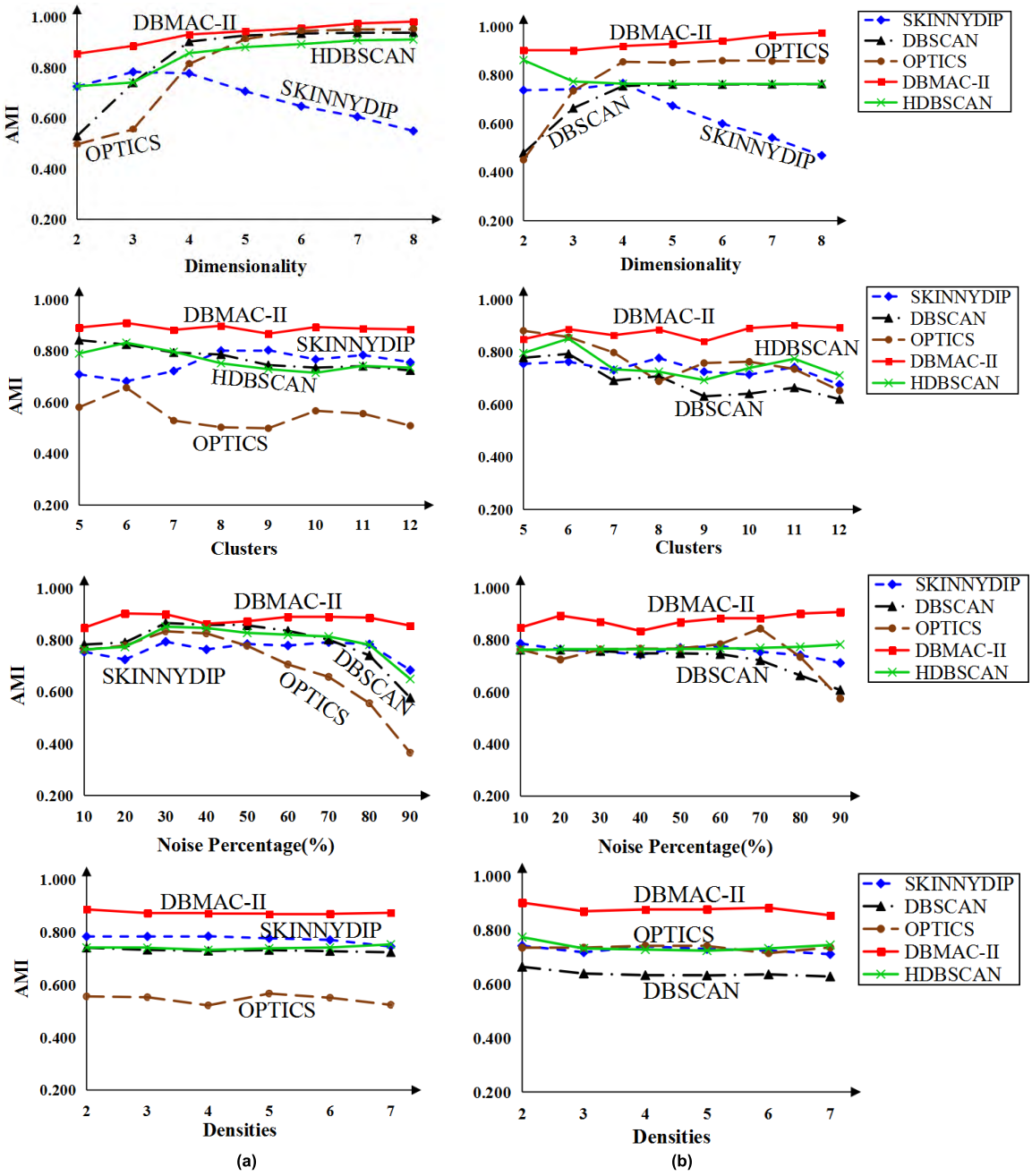

FIGURE 15. Experimental evaluation on multi-dimensional synthetic datasets. (a) AMI with regard to all objects in data. (b) AMI with regard to "clustered" objects in data.

Based on the default parameter values, we systematically varied the dimensionality $d = \{2, 3, \ldots, 8\}$, the number of clusters $k = \{5, 6, \ldots, 12\}$, the noise percentage $\eta = \{10\%, 20\%, \ldots, 90\%\}$ and the number of density levels $l = \{2, 3, \ldots, 7\}$, as shown in Table 2. The parameters settings of each algorithm were the same as in 2D cases.

Figure 15 shows the results of DBMAC-II, DBSCAN, OPTICS, HDBSCAN, and Skinny-dip with different parameter combinations (see Table 2) where the vertical axis indicates the value of AMI. We computed the AMI values with regard to all objects in data (Figure 15(a)) and only the objects that truly belong to a cluster (clustered objects) (Figure 15(b)) for each clustering result respectively. In summary, DBMAC-II shows superior robustness against dimensionality, number of clusters, noise percentage, and density levels, compared to all other four methods in terms of two types of AMI calculation. Note that the performance of DBSCAN and OPTICS was largely dominated by the relative density of clusters compared to noise. For example, as the dimensionality increases, the proportion of space occupied by clusters

TABLE 2. Details of the multi-dimensional synthetic datasets used in Figure 15.

Variable	Fixed Parameters	Details
Dimensionality (1st row)	k=11, η=80%, l=2	Varying the dimensionality d = {2, 3, 4, 5, 6, 7, 8}
Clusters (2nd row)	d=3, η=80%, l=2	Varying the number of clusters k = {5, 6, 7, 8, 9, 10, 11, 12}
Noise Percentage (3rd row)	d=3, k=11, l=2	Varying the noise percentage η = {10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%}
Densities (4th row)	d=3, k=11, η=80%	Changing the number of data points in each cluster to vary the number of density levels l = {2, 3, 4, 5, 6, 7}

will shrink dramatically, increasing the difference in density and leading to better performance of DBSCAN and OPTICS. By contrast, as the number of clusters increases (more ''clustered'' objects), more noise needs to be generated to maintain the same noise percentage, leading to higher noise density and deteriorating performance. Similarly, higher noise percentage will result in higher noise density and worse performance.

V. CONCLUSION

The major motivation of our work is to develop a competent clustering algorithm that can effectively handle datasets with strong noise and highly irregular clusters with varying densities. Based on the principle of the strong noise-robust algorithm DBMAC, we proposed a novel clustering method named DBMAC-II, which is effective at handling clusters of varying-densities in strong noise settings. In DBMAC-II, each data point is featured with its r-neighborhood statistics with different r (radius) values in a way similar to DBMAC. Instead of trying to find a single optimal r value, a set of radius values appropriate for separating ''clustered'' objects and ''noisy'' objects is identified using a formal statistical method for multimodality test, a process referred to as multi-scale analysis. For clusters with different densities, the multiscale analysis process is applied iteratively to extract the ''clustered'' objects with the highest density from the current dataset until the remaining dataset is uniform, which is determined by a statistical test for uniformity. In each iteration, DBSCAN was applied on the extracted ''clustered'' objects to identify the clusters in each level of density. Finally, we put together all clusters discovered from previous iterations as the final result of clustering.

Experiment results show that, compared to classical clustering methods such as DBSCAN, OPTICS, HDBSCAN and the latest strong noise-robust technique Skinny-dip, DBMAC-II features superior effectiveness and robustness in finding arbitrarily shaped clusters with varying densities from datasets with high level noise. In the future, we will investigate the time complexity of DBMAC-II and explore various techniques to improve its scalability. Meanwhile, we will also evaluate DBMAC-II on large scale real-world problems to further validate its performance.

REFERENCES

[1] A. K. Jain, ''Data clustering: 50 years beyond K-means,'' *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.
[2] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data* (Prentice Hall Advanced Reference Series: Computer Science). Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.
[3] D. Pelleg and A. Moore, ''X-means: Extending K-means with efficient estimation of the number of clusters,'' in *Proc. 7th Int. Conf. Mach. Learn.*, 2000, pp. 727–734.
[4] C. B. Do and S. Batzoglou, ''What is the expectation maximization algorithm,'' *Nature Biotechnol.*, vol. 26, no. 8, pp. 897–899, 2008.
[5] L. Zelnik-Manor and P. Perona, ''Self-tuning spectral clustering,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 17, 2004, pp. 1601–1608.
[6] S. Ben-David and N. Haghtalab, ''Clustering in the presence of background noise,'' in *Proc. 31st Int. Conf. Mach. Learn.*, vol. 32, 2014, pp. 280–288.
[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, ''A density-based algorithm for discovering clusters in large spatial databases with noise,'' in *Proc. KDD*, 1996, vol. 96, no. 34, pp. 226–231.
[8] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, ''OPTICS: Ordering points to identify the clustering structure,'' *ACM SIGMOD Rec.*, vol. 28, no. 2, pp. 49–60, Jun. 1999.
[9] R. J. G. B. Campello, D. Moulavi, and J. Sander, ''Density-based clustering based on hierarchical density estimates,'' in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*, vol. 7819, 2013, pp. 160–172.
[10] L. Ertöz, M. Steinbach, and V. Kumar, ''Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data,'' in *Proc. 3rd SIAM Int. Conf. Data Mining*, vol. 112, 2003, pp. 47–58.
[11] T. Pei, A. Jasra, D. J. Hand, A.-X. Zhu, and C. Zhou, ''DECODE: A new method for discovering clusters of different densities in spatial data,'' *Data Mining Knowl. Discovery*, vol. 18, no. 3, pp. 337–369, Jun. 2009.
[12] Y. Zhu, K. M. Ting, and M.-J. Carman, ''Density-ratio based clustering for discovering clusters with varying densities,'' *Pattern Recognit.*, vol. 60, pp. 983–997, Dec. 2016.
[13] T. Zhang, R. Ramakrishnan, and M. Livny, ''BIRCH: An efficient data clustering method for very large databases,'' *ACM SIGMOD Rec.*, vol. 25, pp. 103–114, Jun. 1996.
[14] S. Guha, R. Rastogi, and K. Shim, ''CURE: An efficient clustering algorithm for large databases,'' *ACM SIGMOD Rec.*, vol. 27, no. 2, pp. 73–84, 1998.
[15] C. Böhm, C. Plant, J. Shao, and Q. Yang, ''Clustering by synchronization,'' in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 583–592.
[16] S. Goebl, X. He, C. Plant, and C. Böhm, ''Finding the optimal subspace for clustering,'' in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 130–139.
[17] X. Li, Y. Ye, M. J. Li, and M. K. Ng, ''On cluster tree for nested and multi-density data clustering,'' *Pattern Recognit.*, vol. 43, no. 9, pp. 3130–3143, Sep. 2010.
[18] A. Dasgupta and A. E. Raftery, ''Detecting features in spatial point processes with clutter via model-based clustering,'' *J. Amer. Stat. Assoc.*, vol. 93, no. 441, pp. 294–302, 1998.
[19] W. K. Wong and A. Moore, ''Efficient algorithms for non-parametric clustering with clutter,'' in *Proc. 34th Interface Symp.*, vol. 34, 2002, pp. 541–553.

[20] A. Cuevas, M. Febrero, and R. Fraiman, "Estimating the number of clusters," *Can. J. Stat.*, vol. 28, no. 2, pp. 367–382, 2000.
[21] J. Li, X. Huang, C. Selke, and J. Yong, "A fast algorithm for finding correlation clusters in noise data," in *Proc. 11th Pacific–Asia Conf. Knowl. Discovery Data Mining*, 2007, pp. 639–647.
[22] S. Maurus and C. Plant, "Skinny-dip: Clustering in a sea of noise," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1055–1064.
[23] J. A. Hartigan and P. M. Hartigan, "The dip test of unimodality," *Ann. Stat.*, vol. 13, no. 1, pp. 70–84, 1985.
[24] T. Zhang and B. Yuan, "Density-based multiscale analysis for clustering in strong noise settings," in *Proc. 30th Austral. Joint Conf. Artif. Intell.*, vol. 10400, 2017, pp. 27–38.
[25] L. Kalinichenko, I. Shanin, and I. Taraban, "Methods for anomaly detection: A survey," in *Proc. 16th All-Russian Conf. Digit. Libraries, Adv. Methods Technol., Digit. Collections*, 2014, pp. 20–25.
[26] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," *Very Large Data Bases J.*, vol. 8, nos. 3–4, pp. 237–253, 2000.
[27] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, vol. 29, 2000, pp. 93–104.
[28] E. Schubert, A. Zimek, and H. P. Kriegel, "Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection," *Data Mining Knowl. Discovery*, vol. 28, no. 1, pp. 190–237, 2014.
[29] H. P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Outlier detection in axis-parallel subspaces of high dimensional data," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*, 2009, pp. 831–838.
[30] H. P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Outlier detection in arbitrarily oriented subspaces," in *Proc. 12th Int. Conf. Data Mining*, 2012, pp. 379–388.
[31] S., Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," in *Proc. 19th Int. Conf. Data Eng.*, 2003, pp. 315–326.
[32] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognit. Lett.*, vol. 24, nos. 9–10, pp. 1641–1650, 2003.
[33] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Trans. Knowl. Discovery Data*, vol. 10, no. 1, pp. 51-1–51-5, 2015.
[34] S. P. Smith and A. K. Jain, "Testing for uniformity in multidimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 1, pp. 73–81, Jan. 1984.

TIAN-TIAN ZHANG received the B.E. degree in automation from Central South University, China, in 2015. She is currently pursuing the M.E. degree in control engineering with concentration on data science at Tsinghua University, China.

BO YUAN (S'02–M'07) received the B.E. degree in computer science from the Nanjing University of Science and Technology, China, in 1998, and the M.Sc. and Ph.D. degrees in computer science from The University of Queensland, Australia, in 2002 and 2006, respectively.

From 2006 to 2007, he was a Research Officer with The University of Queensland. Since 2007, he has been with the Graduate School at Shenzhen, Tsinghua University, China, where he is currently an Associate Professor. He has authored over 80 research articles and is the inventor of four patents. His research interests include data analytics, evolutionary computation, and GPU computing.