Research Article

A k -Deviation Density Based Clustering Algorithm

Chen Jungan ^{1,2}, Chen Jinyin ¹, Yang Dongyong ¹ and Li Jun²

¹College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China

²College of Electronic Information, Zhejiang Wanli University, Ningbo 315100, China

Correspondence should be addressed to Yang Dongyong; yangdy@zjut.edu.cn

Received 2 October 2017; Revised 29 December 2017; Accepted 17 January 2018; Published 26 February 2018

Academic Editor: Erik Cuevas

Copyright © 2018 Chen Jungan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the adoption of global parameters, DBSCAN fails to identify clusters with different and varied densities. To solve the problem, this paper extends DBSCAN by exploiting a new density definition and proposes a novel algorithm called k -deviation density based DBSCAN (kDDBSCAN). Various datasets containing clusters with arbitrary shapes and different or varied densities are used to demonstrate the performance and investigate the feasibility and practicality of kDDBSCAN. The results show that kDDBSCAN performs better than DBSCAN.

1. Introduction

DBSCAN is a classical density based clustering method [1] and has many desirable features including good robustness to noise and outliers. However, due to the adoption of global parameters, especially the introduction of neighborhood radius Eps, DBSCAN fails to identify clusters with different and varied densities. To solve this problem, two main methods have been proposed as follows.

(1) *Adaptive Local Density or Eps.* GRIDBSCAN [2] and GMDDBSCAN [3] use the grid technique to calculate the local density (Eps, MinPts), where MinPts is defined as the minimum neighbors of a point when considering the point as the core point. APSCAN [4] uses the Affinity Propagation (AP) algorithm to partition a dataset into some patches and calculate the local density of each patch. VDBSCAN [5] uses a k -dist plot to select several Eps values for different densities. Multi-DBSCAN [6] uses the must-link constraint and k -nearest distance to calculate Eps values for different densities. DBSCAN-DLP [7] partitions a dataset into many subsets with different density levels by analyzing the statistical characteristics of its density variation and then estimates the Eps value for each subset. DSets-DBSCAN regards the data in the dominant set as core points and those from extrapolation as border ones, so Eps can be determined automatically based on the dominant set [8]. After the local density or Eps

is estimated, all these algorithms apply DBSCAN to merge those data with similar density.

EDBSCAN [9] assigns varied values for Eps according to the local density based on the k -nearest neighbors, and the clustering process starts from the highest local density point towards the lowest local density one. DDSC [10] uses the Homogeneity test to detect the density difference between different regions; if their density difference is less than α , those regions will be merged into the same cluster.

(2) *Redefinition of the Density with No Parameter Eps.* H-density [11] estimates the local density of the nonnormalized probability distribution according to the neighborhood of radius R , and the hierarchical agglomerative strategy is used to merge clusters according to the di-similarity measures. In the multidensity DBSCAN, two adjacent spatial regions are separated into two clusters when the difference between DST and AVGDST violates a threshold, where DST is the average distance between one point and its k -nearest neighbors and AVGDST is the average distance between any point in one cluster and its k neighbors [12]. In K -DBSCAN [13], the K -means clustering algorithm is employed to divide all points into K -level groups based on their l -density values (here, l -density value is the average distance of the point P and its l -nearest neighbors), and then DBSCAN is used to merge similar data according to the density levels.

FIGURE 1: Dataset with varied densities.

Among these methods, Eps is automatically calculated according to different densities in the first method, and the definitions of different densities are proposed in the second method. Hereinto, one kind of definition is based on the k -nearest neighborhood method such as the local density of the nonnormalized probability distribution [11], the average distance between one point and its k -nearest neighbors [12], l -density [13], and k neighborhood density [7]. Based on these definitions, the varied densities can be represented separately by di-similarity measures [11], the difference between DST and AVGDST [12], and the density variation or density level [7].

It is known that the main objective of defining the density concept is to cluster the objects with similar density into the same cluster. For example, in Figure 1, the blue circles of A, B, C, and D can be viewed as one normal cluster, while the red circle E is an abnormal one. However, it is difficult to describe the density with Eps and MinPts because of different and varied densities. For example, in Figure 1(a), the fixed Eps have different densities in all these circle regions and the difference between circle E and another circle is not apparent.

Inspired by the above discussion, redefinition of the density based on the k -nearest neighbors seems a better solution to solve the problem of different and varied densities. Unfortunately, the density based on the average distance between one point and its k -nearest neighbors [7, 12, 13] is not good enough to describe the varied densities. For example, in Figure 1(b), if $k = 7$, the average distance is different in all these circle regions and circle E cannot be distinguished from other circles.

In this work, a new density definition called k -deviation density is proposed to describe the different and varied densities. The k -deviation density is defined as the proportion between the maximum distance and average distance. Given $k = 7$ in Figure 1(b), the densities of the circle regions of A, B, C, and D are similar, while the density of the circle region E is different from that of A, B, C, or D. Thus a k -deviation density based clustering algorithm (kDDBSCAN) can be put forward.

2. The kDDBSCAN Algorithm

The basic idea of this paper is that the objects in the same cluster have the similar and small k -deviation density which can reflect the deviation of an object from others. Based on the k -deviation density and DBSCAN, kDDBSCAN is proposed. In this section, some basic concepts or definitions

are given, and the process of the proposed algorithm is described in detail.

The points used to calculate the k -deviation are sampled through the k -nearest neighborhood (KNN) method. According to [14], the shared nearest neighborhood (SNN) method can be used due to its robustness in high dimension dataset. However, SNN is not efficient because of its complexity. In view of this, the k neighborhood method is proposed to combine KNN and SNN.

When k points are sampled through the k neighborhood method, the k -deviation density can be calculated. The smaller k -deviation density means that it is more likely that these k samples are in the same cluster. Hence, a deviation factor is proposed as the given threshold to decide whether the given datasets are in the same cluster. If the k -deviation is greater than the deviation factor, the given datasets are not in the same cluster. This process is called directly density-reachable (DDR) and can be used to identify core points. Furthermore, density-reachable is proposed to decide whether two core points belong to the same cluster.

2.1. Basic Definitions

Definition 1 (k -nearest neighborhood). The k -nearest neighborhood (KNN) of a point x is denoted by $N_k(x)$.

Definition 2 (mutual k -nearest neighborhood). Given two points x_i and x_j , if $x_i \in N_k(x_j)$ and $x_j \in N_k(x_i)$, then x_i and x_j are mutual k -nearest neighbors. The mutual k -nearest neighborhood (mKNN) of a point x is denoted by $M_k(x)$.

Definition 3 (k neighborhood). The k neighborhood of a point x is denoted by $NM_k(x, n)$, where n is the minimal number of the mutual neighborhood:

$$\begin{aligned} NM_k(x, 0) &= N_k(x) \quad n = 0 \\ NM_k(x, n) &= M_k(x) \bigwedge |M_k(x)| \geq n \quad n > 0. \end{aligned} \quad (1)$$

Definition 4 (k -deviation density). Let $x_i \in NM_k(x, n)$ and $d(x, x_i)$ be the distance from a point x to its i th nearest neighbor. Then, the k -deviation density is defined as

$$\text{Dev}_k(x, n) = \frac{\max_{x_i \in NM_k(x, n)} (d(x, x_i))}{\text{avg}_{i \in \text{argmax}(d(x, x_i))} (d(x, x_i))}. \quad (2)$$

Definition 5 (directly density-reachable). A point p is directly density-reachable from a point q if $p \in NM_k(q, n)$ and

Input: DataSt D , k , the minimal number of the mutual neighborhood n , the deviation factor α
Procedure kDDBSCAN(D, k, n, α)
 (1) $C = 0$ /* C is cluster id */
 (2) For each point P_i in D do
 (3) If $P_i \cdot \text{Cid} = \text{UNCLASSIFIED}$ Then
 (4) Calculate the k -deviation density $\text{Dev}_k(P_i, n)$ /* see Definition 4 */
 (5) If $\text{Dev}_k(P_i, n) \leq \alpha$ Then
 (6) **ExpandCluster**(P_i, C)
 (7) $C = C + 1$
 (8) Else
 (9) $P_i \cdot \text{Cid} = \text{OUTLINE}$
 (10) End If
 (11) End For
 (12) End For
 End Procedure

ALGORITHM 1: The implementation of kDDBSCAN.

Procedure ExpandCluster(P_0, C)
 (1) $\text{CorePoints} = \text{CorePoints} \cup P_0$
 (2) For each point P_i in CorePoints do
 (3) $P_i \cdot \text{Cid} = C$ /* Assign P_i to Cluster C */
 (4) For each point P_j in $\text{NM}_k(P_i, n)$ do
 (5) If $P_j \cdot \text{Cid} = \text{UNCLASSIFIED} \parallel \text{OUTLINE}$ Then
 (6) $P_j \cdot \text{Cid} = C$ /* Assign P_j to Cluster C */
 (7) End If
 (8) Calculate k -deviation density $\text{Dev}_k(P_j, n)$ /* see Definition 4 */
 (9) If $\text{Dev}_k(P_j, n) \leq \alpha$ & P_i and P_j is density-reachable Then /* see Definition 6 */
 (10) $\text{CorePoints} = \text{CorePoints} \cup P_j$
 (11) Else
 (12) $P_j \cdot \text{Cid} = \text{OUTLINE}$
 (13) End If
 (14) End For
 (15) End For
 End Procedure

ALGORITHM 2: The procedure of ExpandCluster.

$\text{Dev}_k(x) \leq \alpha$ (core point condition), where α is the deviation factor.

Definition 6 (density-reachable). A point p is density-reachable from a point q if the following conditions are satisfied:

- (1) $\max(\max_{x_i \in \text{NM}_k(x, n)}(d(p, x_i)) / \max_{x_i \in \text{NM}_k(x, n)}(d(q, x_i)), \max_{x_i \in \text{NM}_k(x, n)}(d(q, x_i)) / \max_{x_i \in \text{NM}_k(x, n)}(d(p, x_i))) \leq \alpha$
- (2) $\max(\text{avg}_{x_i \in \text{NM}_k(x, n)}(d(p, x_i)) / \text{avg}_{x_i \in \text{NM}_k(x, n)}(d(q, x_i)), \text{avg}_{x_i \in \text{NM}_k(x, n)}(d(q, x_i)) / \text{avg}_{x_i \in \text{NM}_k(x, n)}(d(p, x_i))) \leq \alpha$
- (3) $\max(\max_{x_i \in \text{NM}_k(x, n)}(d(p, x_i)) / \text{avg}_{x_i \in \text{NM}_k(x, n)}(d(q, x_i)), \max_{x_i \in \text{NM}_k(x, n)}(d(q, x_i)) / \text{avg}_{x_i \in \text{NM}_k(x, n)}(d(p, x_i))) \leq \alpha$
- (4) $\max(d(p, q) / \max_{x_i \in \text{NM}_k(x, n)}(d(p, x_i)), \max_{x_i \in \text{NM}_k(x, n)}(d(p, x_i)) / d(p, q)) \leq \alpha$
- (5) $\max(d(p, q) / \max_{x_i \in \text{NM}_k(x, n)}(d(q, x_i)), \max_{x_i \in \text{NM}_k(x, n)}(d(q, x_i)) / d(p, q)) \leq \alpha$

2.2. The Process of kDDBSCAN. The procedure of ExpandCluster in Algorithm 1 is shown in Algorithm 2. The proposed kDDBSCAN, which is an extension of DBSCAN, is depicted as Algorithm 1.

3. Experiments

To thoroughly evaluate the effectiveness of kDDBSCAN, this section covers various types of datasets including clusters with arbitrary shape, uniform density, or varied densities. Since clusters in 2D datasets are easy to be visualized and compared by different algorithms, the performance comparison of kDDBSCAN with DBSCAN and the evaluation on parameter sensitivity are firstly conducted in the two-dimensional space. Then we demonstrate that the proposed algorithm is applicable to multidimensional datasets. UCI

FIGURE 2: Spiral dataset.

datasets with ground truth are used to quantify the performance according to the clustering results from the perspectives of Homogeneity, Completeness, V -measure, ARI, FMI, and NMI. Finally, BSDS500, Olivetti Face dataset, and MNIST are used to investigate the feasibility and practicality of kDDBSCAN. In kDDBSCAN, the default values are $n = 1$ and $\alpha = 999999$.

3.1. Clustering on the Two-Dimensional Dataset. Two-dimensional datasets are chosen according to their different characteristics. Spiral dataset represents the dataset with well-separated and nonspherical cluster. Aggregation and Flame datasets have adjacent regions with uniform density. D1 and Path based datasets represent the dataset containing embedded and adjacent clusters with different densities. Jain dataset contains sparse data regions with different densities. Compound dataset contains adjacent, embedded regions with varied and different densities.

3.1.1. Comparison with DBSCAN. Figures 2–8 show the results from DBSCAN and kDDBSCAN on those two-dimensional datasets, respectively, and the corresponding parameters for each algorithm on each kind of dataset are also given. Since the k neighborhood method is different for $n = 0$ and $n > 0$, the results when $n = 0$ are given separately. When the parameters n and α are set as default values, kDDBSCAN has only one parameter k .

(1) Spiral dataset contains well-separated and nonspherical cluster. Figure 2 shows that both DBSCAN and kDDBSCAN can obtain correct results.

(2) Aggregation and Flame datasets contain adjacent regions with uniform density. Figure 4 shows that the

KNN method performs worse than DBSCAN or mKNN method.

(3) D1 and Path based datasets contain embedded and adjacent clusters with different densities.

In Figure 5, kDDBSCAN performs better than DBSCAN because the k -deviation density has considered the deviation factor of the dataset with different densities. Figure 6 shows that mKNN performs better than DBSCAN and KNN, indicating that mKNN is effective for adjacent clusters.

(4) Jain dataset contains sparse data regions with different densities.

In Figure 7, DBSCAN cannot obtain correct results because of data sparsity, whereas kDDBSCAN performs well because the k -deviation density has considered the deviation factor of the dataset with sparse data.

(5) Compound dataset contains adjacent, embedded regions with varied and different densities. In Figure 8, DBSCAN takes the sparse data as noise while kDDBSCAN does not.

Through the above discussion, it can be seen that the k -deviation density can handle the dataset with varied densities or sparse data. The parameter α is effective in clustering the adjacent region according to the k -deviation density, as illustrated in Figures 3–5. The method of mKNN ($n = 1$) has the ability to cluster the well-separated data, as illustrated in Figures 2, 7, and 8.

3.1.2. Parameter Sensitivity. In Figure 9, k is sensitive to different densities in sparse data regions as the k value decides the nearest neighborhood.

In Figure 10, n is sensitive to varied and different densities when adjacent regions exist.

FIGURE 3: Aggregation dataset.

FIGURE 4: Flame dataset.

In Figure 11, as the parameter α is effective in clustering the adjacent regions according to the k -deviation density, it is sensitive when clustering adjacent regions. However, n is not sensitive to uniform density.

3.2. Clustering on Multidimensional Datasets. To demonstrate the applicability of kDDBSCAN to multidimensional

dataset, UCI datasets with ground truth are used to evaluate its performance according to the clustering results from the perspectives of Homogeneity, Completeness, V-measure, ARI, FMI, and NMI. The input parameters are listed as Table 1, and the corresponding results are shown in Table 2. As can be seen, both algorithms can get the same results with Iris, but kDDBSCAN is more effective for other datasets.

FIGURE 5: D1 dataset.

FIGURE 6: Path based dataset.

3.3. Application of kDDBSCAN to Image Segmentation. Berkeley Segmentation Data Set and Benchmarks 500 (BSDS500) consist of 500 natural images with ground-truth human annotations [15]. Among these images, three are used to demonstrate the effectiveness of kDDBSCAN relative to DBSCAN, and the results are shown in Table 3 and Figure 12. Here, the data point is composed of the pixel location and

the corresponding RGB value. In addition, the parameter sensitivity of kDDBSCAN is analyzed in this section and the results are shown in Figures 13–18.

3.3.1. Comparison with DBSCAN. As shown in Figure 12, the segmentation and boundary detection is basically matched with the ground truth although some noise is required to

FIGURE 7: Jain dataset.

FIGURE 8: Compound dataset.

TABLE 1: The input parameters of different datasets for DBSCAN and kDDBSCAN.

	Iris	Glass	Wdbc	Segment
DBSCAN	(1.6, 15)	(1.2, 5)	(2.3, 5.0)	(2.0, 10)
kDDBSCAN	(20, default, default)	(12, default, default)	(7, default, default)	(8, default, default)

FIGURE 9: Jain dataset.

TABLE 2: The corresponding results from DBSCAN and kDDBSCAN.

		Iris	Glass	Wdbc	Segment
Homogeneity	DBSCAN	0.579	0.306	0.305	0.416
	kDDBSCAN	0.579	0.361	0.385	0.451
Completeness	DBSCAN	1	0.502	0.220	0.582
	kDDBSCAN	1	0.540	0.336	0.589
V-measure	DBSCAN	0.733	0.380	0.256	0.485
	kDDBSCAN	0.733	0.433	0.359	0.511
ARI	DBSCAN	0.568	0.245	0.230	0.222
	kDDBSCAN	0.568	0.289	0.409	0.247
FMI	DBSCAN	0.771	0.527	0.607	0.430
	kDDBSCAN	0.771	0.558	0.743	0.445
NMI	DBSCAN	0.761	0.392	0.259	0.492
	kDDBSCAN	0.761	0.442	0.360	0.515

TABLE 3: The corresponding results with BSDS500 from DBSCAN and kDDBSCAN.

Index	Algorithm	Homogeneity	Completeness	V-measure	ARI	FMI	NMI
35070	kDDBSCAN	0.775	0.409	0.536	0.598	0.787	0.563
	DBSCAN	0.591	0.431	0.498	0.507	0.729	0.504
35008	kDDBSCAN	0.749	0.417	0.536	0.687	0.883	0.559
	DBSCAN	0.535	0.597	0.564	0.707	0.915	0.565
22090	kDDBSCAN	0.738	0.626	0.677	0.765	0.808	0.680
	DBSCAN	0.575	0.612	0.593	0.550	0.655	0.594

FIGURE 10: Compound dataset.

be further processed. In kDDBSCAN, more noise can be found. Nevertheless, kDDBSCAN can separate the flower of BSD35008 and the hill of BSD22090 more clearly than DBSCAN does. The results in Table 3 indicate that kDDBSCAN can achieve better cluster performance than DBSCAN. In particular, the values of Homogeneity and FMI for kDDBSCAN are both above 0.7.

3.3.2. Parameter Sensitivity of kDDBSCAN. In this part, the visual image segmentation results by kDDBSCAN are

discussed, and then the clustering results with different parameter values using kDDBSCAN are evaluated.

(1) Results of the Segmentation with Different Parameter Values. In Figure 13, with the increase of α value, the likelihood that similar pixels are clustered into the same cluster also increases, because higher α value means larger tolerance to the difference between pixels. Similarly, in Figure 14, with the increase of n value, similar pixels have less likelihood to be clustered into the same cluster. In general, higher α value and

FIGURE 11: Flame dataset.

lower n value lead to larger likelihood for similar pixels to be clustered into the same cluster.

As the parameter k decides the number of sampled points, the deviation density of every core point does not vary monotonically when k increases. Consequently, there must be multiple optimum k values to achieve better performance when α and n are constant. This is in accordance with the different results of the last images with different k values in Figure 15.

(2) *Clustering Evaluation.* Figures 16–18 show the results with different parameter values using kDDBSCAN for different evaluation indexes. Based on the analysis of Figures 13 and 14,

higher α value and lower n value can lead to larger likelihood for similar pixels to be clustered into the same cluster. As can be seen from Figures 16 and 17, most evaluation indexes vary monotonically with α and n . By contrast, in Figure 18, the Homogeneity value does not vary monotonically with k , so there is an optimum value k required to be set manually.

3.4. Application of kDDBSCAN to Olivetti Face Dataset and MNIST. kDDBSCAN and DBSCAN are applied to the Face dataset to group the images for the same person to a cluster without any previous training. Olivetti Face dataset is a widespread benchmark for machine learning algorithms. There are ten different images for each of 40 distinct persons.

FIGURE 12: Image segmentation results with BSDS500 from kDDBSCAN and DBSCAN (the images in the first row are the original images and ground-truth images, and other images are the segmentation and boundary detection results).

FIGURE 13: Results of the segmentation with different α values ($k = 12, n = 1$).

FIGURE 14: Results of the segmentation with different n values ($k = 10, \alpha = 1.5$). FIGURE 15: Results of the segmentation with different k values ($n = 1, \alpha = 1.5$). FIGURE 16: Results with different α values using kDDBSCAN ($k = 12, n = 1$). FIGURE 17: Results with different n values using kDDBSCAN ($k = 10, \alpha = 1.5$).

FIGURE 18: Results with different k values using kDDBSAN ($n = 1, \alpha = 1.5$).

FIGURE 19: DBSCAN (4, 0.13) with Olivetti Face dataset.

Similar to [16], the similarity between two images is calculated according to the method in [17]. Figures 19 and 20 show the clustering results, where the images of the same color correspond to the same cluster and the images with red point are taken as noise. It can be seen that DBSCAN identifies six persons as the same person “1,” whereas kDDBSAN only identifies two persons as the same person “4.” The values of those clustering evaluation indexes such as Homogeneity, Completeness, V-measure, ARI, FMI, and NMI are given in

Table 4. It is clear that kDDBSAN performs better than DBSCAN.

The MNIST database of handwritten digits has a training set of 60,000 examples and a testing set of 10,000 examples. The digits have been size-normalized and centered in a fixed-size image [18]. 500 examples of every digit are randomly chosen and put into kDDBSAN. Table 5 provides the results of clustering evaluation, and Figure 21 displays the visual clustering results. As can be seen from Table 5, the maximum

FIGURE 20: kDDBSCAN (4, default, default) with Olivetti Face dataset.

TABLE 4: The corresponding results of clustering evaluation with Olivetti Face dataset.

Index	Homogeneity	Completeness	V-measure	ARI	FMI	NMI
kDDBSCAN (4, default, default)	0.818	0.787	0.802	0.434	0.449	0.802
DBSCAN (4, 0.13)	0.502	0.775	0.609	0.095	0.216	0.624

TABLE 5: The corresponding results of clustering evaluation with MNIST dataset.

K	n	Alpha	Homogeneity	Completeness	V-measure	ARI	FMI	NMI
10	3	1.15	0.720	0.374	0.492	0.096	0.173	0.519
		1.17	0.700	0.397	0.506	0.147	0.22	0.527
		1.2	0.608	0.45	0.517	0.261	0.337	0.523
12	4	1.15	0.719	0.375	0.493	0.087	0.165	0.519
		1.17	0.698	0.405	0.513	0.153	0.228	0.532
		1.2	0.649	0.476	0.549	0.294	0.366	0.556
15	4	1.15	0.582	0.469	0.519	0.281	0.376	0.522
		1.17	0.561	0.532	0.546	0.323	0.427	0.546
		1.2	0.288	0.471	0.358	0.094	0.317	0.368

Homogeneity can reach 0.72, whereas the maximum ARI is only 0.323. The higher Homogeneity value in Table 5 means the larger maximum number of the clusters in the top-left of Figure 21, while the higher ARI value means that more similar examples can be clustered into the same cluster. Obviously, these two clustering indexes are contradictory. Hence, which clustering index is better depends on the actual application circumstance. From Figure 21, it can be concluded that the digits (4, 7, 9) and (5, 8) are easy to be grouped into the same cluster.

4. Conclusions

In summary, the basic idea of this paper is that the objects in the same cluster have the similar and small k -deviation density which can reflect the deviation of an object from others. On this basis, kDDBSCAN is proposed based on the

k -deviation density and DBSCAN to identify clusters with different and varied densities, and various datasets containing clusters with arbitrary shapes, uniform density, and different or varied densities are used to demonstrate the performance of kDDBSCAN. The results show that kDDBSCAN can achieve better results than DBSCAN.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61502423), Zhejiang Provincial Natural Science Foundation (nos. Y14F020092,

FIGURE 21: kDBSCAN with MNIST dataset (in each figure, the blue line represents the ground truth, the red line represents the clustering index value).

LY14F040002, and LY16G020012), Zhejiang Public Welfare Technology Research Project Foundation (no. 2017C31040), and Ningbo Natural Science Foundation (nos. 2015A610130 and 2013A610006).

References

- [1] M. Ester, H. P. Krigel, J. Sander, and X. Xu, "A Density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.
- [2] O. Uncu, W. A. Gruver, D. B. Kotak, D. Sabaz, Z. Alibhai, and C. Ng, "GRIDBSCAN: GRID density-based spatial clustering of applications with noise," in *Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics*, pp. 2976–2981, Taiwan, October 2006.
- [3] C. Xiaoyun, M. Yufang, Z. Yan, and W. Ping, "GMDBSCAN: Multi-density DBSCAN cluster based on grid," in *Proceedings of the IEEE International Conference on e-Business Engineering, ICEBE'08*, pp. 780–783, China, October 2008.
- [4] X. Chen, W. Liu, H. Qiu, and J. Lai, "APSCAN: A parameter free algorithm for clustering," *Pattern Recognition Letters*, vol. 32, no. 7, pp. 973–986, 2011.
- [5] L. Peng, Z. Dong, and W. Naijun, "VDBSCAN: Varied Density Based Spatial Clustering of Applications with Noise," in *Proceedings of the ICSSSM'07: 2007 International Conference on Service Systems and Service Management*, China, June 2007.
- [6] T.-Q. Huang, Y.-Q. Yu, K. Li, and W.-F. Zeng, "Reckon the parameter of DBSCAN for multi-density data sets with constraints," in *Proceedings of the 2009 International Conference on Artificial Intelligence and Computational Intelligence, AICI 2009*, pp. 375–379, China, November 2009.
- [7] Z. Xiong, R. Chen, Y. Zhang, and X. Zhang, "Multi-density DBSCAN algorithm based on density levels partitioning," *Journal of Information and Computational Science*, vol. 9, no. 10, pp. 2739–2749, 2012.
- [8] J. Hou, H. Gao, and X. Li, "DSets-DBSCAN: a parameter-free clustering algorithm," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3182–3193, 2016.
- [9] A. Ram, A. Sharma, A. S. Jalal, R. Singh, and A. Agrawal, "An enhanced density based spatial clustering of applications with noise," in *Proceedings of the 2009 IEEE International Advance Computing Conference, IACC 2009*, pp. 1475–1478, India, March 2009.
- [10] B. Borah and D. K. Bhattacharyya, "DDSC: A density differentiated spatial clustering technique," *Journal of Computers*, vol. 3, no. 2, pp. 72–79, 2008.
- [11] D. Pascual, F. Pla, and J. Sanchez, "Non parametric local density-based clustering for multimodal overlapping distributions," *Intelligent Data Engineering and Automated Learning IDEAL*, pp. 671–678, 2006.
- [12] W. Ashour and S. Sunoallah, "Multi density DBSCAN," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6936, pp. 446–453, 2011.
- [13] M. Debnath, P. K. Tripathi, and R. Elmasri, "K-DBSCAN: Identifying spatial clusters with differing density levels," in *Proceedings of the 2015 International Workshop on Data Mining with Industrial Applications, DMIA 2015*, pp. 51–60, Paraguay, September 2015.
- [14] M. E. Houle, H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Can shared-neighbor distances defeat the curse of dimensionality?" *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6187, pp. 482–500, 2010.
- [15] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the 8th International Conference on Computer Vision*, pp. 416–423, July 2001.
- [16] A. Laio and A. Rodriguez, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.
- [17] M. P. Sampat, Z. Wang, S. Gupta, A. C. Bovik, and M. K. Markey, "Complex wavelet structural similarity: a new image similarity index," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2385–2401, 2009.
- [18] Y. LeCun and C. Cortes, "The mnist database of handwritten digits," Tech. Rep., Available electronically at <http://yann.lecun.com/exdb/mnist>, 2012.