2021 IEEE International Conference on Data Mining (ICDM)

2021 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-2398-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICDM51629.2021.00158

Density-Based Clustering for Adaptive Density Variation

Li Qian*, Claudia Plant†, Christian Böhm*
*Institute of Informatics, Ludwig Maximilian University of Munich, Munich, Germany
{li.qian, boehm}@dbs.ifi.lmu.de
†Faculty of Computer Science, ds:UniVie, University of Vienna, Vienna, Austria
claudia.plant@univie.ac.at

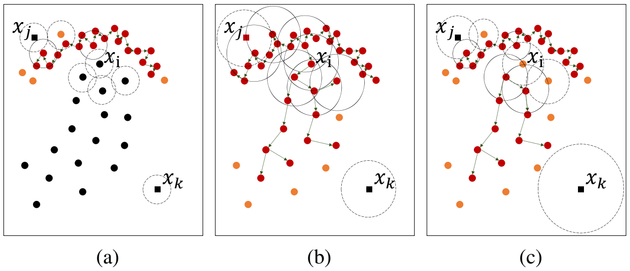
***Abstract*—Cluster analysis plays a crucial role in data mining and knowledge discovery. Although many researchers have investigated clustering algorithms over the past few decades, most of the well-known algorithms have shortcomings when dealing with clusters of arbitrary shapes and varying sizes and in the presence of noise and outliers. Density-based methods partially solve these issues but fail to discover clusters with varying densities. In this paper, we propose a novel Density-Based clustering algorithm for Adaptive Density Variation (DBADV), which is based on the classic clustering algorithm DBSCAN. To address the problem of density variation, we define the local density information, which not only reflects the individual property of each object but also describes the density distribution of clusters, and finds the adaptive search range of each object by collecting information from its neighbors. Moreover, we design a new metric to obtain the mutual nearest neighbors of each object to better detect the objects around the boundaries between clusters. We show the effectiveness of our method in extensive experiments on synthetic and real-world data sets, which demonstrate that the performance of the proposed algorithm DBADV is superior to other competitive clustering algorithms.**

***Keywords*-density-based clustering; adaptive density variation; mutual nearest neighbor search**

I. Introduction

As one of the most important unsupervised learning techniques, clustering algorithms are required to be applied in various data analysis fields. Clustering aims to find natural groupings of data, such that objects within the same cluster are similar to each other while objects in different clusters are dissimilar. However, the existing clustering algorithms have some limitations: they cannot handle clusters of arbitrary shapes and sizes, are sensitive to noise and outliers, require prior knowledge (e.g., predefined number of clusters), and cannot maintain a relatively reasonable computational complexity.

To tackle the challenges above, we aim to define the local density information, which contributes to describing the distribution of clusters with fine granularity. Considering that density-based methods focus on density information but fail to deal with clusters with varying densities, we propose a novel Density-Based clustering algorithm for Adaptive Density Variation (DBADV) that inherits such density property and attempts to use the local density information to solve the problem of density variation. As the classic representative of density-based methods, the basic idea of DBSCAN [1] is to use a global search range (ε-*neighborhood*) for all objects to distinguish relatively dense regions from sparse regions. In contrast, DBADV regards the local density information as the individual property of each object. Namely, each object has its own adaptive search range. A toy data set can illustrate the intuition of our method to shed some light on the difference between DBSCAN and DBADV. As shown in Fig. 1, the toy data set has one upper half-moon cluster with high density, one random-shaped cluster with low density, and two distinct outliers marked as squares. x_i, x_j, and x_k represent boundary point and outliers, respectively. We set the minimum number of points $MinPts$ within the search range to 4 (including itself). DBSCAN and DBADV treat points containing at least $MinPts$ within their search range as *core points* (red). The points that are not *core points* but within the search range of *core points* are called *border points* (orange). The remaining points after clustering are *outliers* (black). The green arrow denotes the search path from the current *core point* to the next *core point*, the solid circle represents the search range of the *core point*, and the dashed circle represents the search range of the non-core point. For better display, we only highlight partial circles of the representative points. Fig. 1(a) shows that DBSCAN, with a small global search range, can find the dense upper half-moon but assigns all points in the sparse region as *outliers*. For a large global search range, DBSCAN treats the boundary point x_i as the next *core point* and further merges



(a) (b) (c)

Figure 1. The clustering process of DBSCAN and DBADV with different search range (partial). (a) DBSCAN with a small global search range; (b) DBSCAN with a large global search range; (c) DBADV with the adaptive search range.

978-1-6654-2398-4/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDM51629.2021.00158

all points in the sparse region to one cluster, including the *outlier* x_j, as shown in Fig. 1(b). On the contrary, DBADV finds the adaptive search range of each point through the local density information collected from neighbors. The adaptive search range is small in the relatively dense region and large in the sparse region. Since each point has its own adaptive search range, DBADV correctly identifies two clusters with different densities and two outliers, as shown in Fig. 1(c).

The paper makes the following contributions:

- We define the local density information and find the adaptive search range for each object, thus enabling DBADV to discover clusters with varying densities.
- We design a new metric to search the mutual nearest neighbor of each object, which can better detect the objects around the indistinguishable boundaries between clusters.
- The experimental results show that DBADV can handle clusters of arbitrary shapes and sizes with varying densities, and is robust to noise and outliers.

II. Related Work

Density-based methods [2] can effectively exploit density information to find clusters of arbitrary sizes and shapes while remaining robust against noise and outliers, and do not require the user to specify the number of clusters. However, it fails to identify clusters with varying densities due to the global search range, such as DBSCAN. Abundant improvements of DBSCAN have attempted to overcome this limitation. OPTICS [3] defines the *reachability distance* and draws a reachability plot, such that all points are sorted in a special linear order, with spatially adjacent points following each other closely. OPTICS relies on order points to identify the clustering structure and expects some kind of density drop to detect cluster borders. DScale [4] is an adaptive multi-dimensional scaling method that simultaneously considers all dimensions to rescale a given data set, and then applies the rescaled data set to an existing density-based method, such as DBSCAN. DScale as a preprocessing can improve accuracy but requires an extensive search of three parameters. Clustering with Robust Autocuts and Depth (CRAD) [5] is based on a new neighbor searching function by using a notion of statistical data depth as the dissimilarity measure. CRAD is restricted by its algorithm design (e.g., requiring invertible matrices), thus cannot process data in general.

From another perspective, some approaches combine the benefits of density-based methods and other categories of clustering algorithms while alleviating some of the inherent disadvantages of each algorithm. The Hierarchical DBSCAN (HDBSCAN) [6] combines density-based and hierarchical-based methods. HDBSCAN forms a Minimum Spanning Tree (MST), connecting all points in the hyperspace and defines *mutual reachability distance* as its edge weight between two vertexes in MST. However, the processing of boundary points for HDBSCAN is not ideal, and the minimum number of clusters needs to be defined. SpectACl [7] combines density-based and graph-based methods. SpectACl uses minimum cut from Spectral clustering [8] and maximum density from DBSCAN to find clusters with a large average density. The appropriate density for each cluster is automatically determined through the spectrum of the weighted adjacency matrix. Nevertheless, SpectACl requires the user to specify the number of clusters and cannot handle outliers.

Recently, some clustering algorithms based on the Dip-Test for multimodality have been proposed [9, 10]. These algorithms are robust against outliers but are restricted to unimodal clusters. Synchronous (Sync) [11–13] is a parameter-free clustering algorithm based on the Kuramoto model to simulate the dynamics of each point during the process toward synchronization and discover clusters and outliers automatically. The spectral clustering algorithm [14, 15] is very popular because it can detect arbitrary-shape clusters in data spectrum space. However, spectral clustering has high complexity, cannot handle outliers, and requires the number of clusters.

III. Proposed Method

A. Local Density Information

In information theory, perplexity is used as a measurement to evaluate distribution or model performance, such as in language models [16]. Recently, the notion of perplexity has been applied to other fields. t-Distributed Stochastic Neighbor Embedding (t-SNE) [17], an effective technique for dimensionality reduction, uses perplexity to find the bandwidth of each object. Inspired by that, we aim to find such information as the local density information reflecting individual property for each object through perplexity. To the best of our knowledge, this is the first time to apply the notion of perplexity to clustering algorithms.

The perplexity of a discrete probability distribution p is defined as

$$Perplexity(p) = 2^{H(p)}, \tag{1}$$

where $H(p)$ is the Shannon entropy of p measured in bits: $H(p) = -\sum_{x} p(x)\log_2 p(x)$.

For a data set $D = \{x_1, ..., x_n\}$, we define the local density information of a point x_i by collecting all the conditional probability $p_{j|i}$ with a neighbor point x_j. This conditional probability shows the similarity between two points. For a close point, $p_{j|i}$ is relatively high, whereas, for a faraway point, $p_{j|i}$ is almost infinitesimal. Mathematically, we define this conditional probability $p_{j|i}$ as

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{l \neq i} \exp\left(-\|x_i - x_l\|^2 / 2\sigma_i^2\right)}, \quad (2)$$

where σ_i is the bandwidth of the Gaussian kernel centered on x_i, and x_l is any point in the data set except x_i. The bandwidth of a point in the dense region is usually smaller than that in the sparse region. Therefore, we can consider the bandwidth of each point as the local density information. The local density information not only reflects the individual property of each object but also describes the density distribution of clusters with fine granularity.

We set the same perplexity for the probability distribution of each point in the entire data set, which means that the probability distribution of each point has the same Shannon entropy $H(p_i) = -\sum\limits_j p_{j|i} \log_2 p_{j|i}$. Consequently, a binary search is performed to find the bandwidth of each point by approximating the Shannon entropy of the conditional probability p_i to the logarithm of the fixed perplexity.

B. Adaptive Search Range

Unlike DBSCAN, which performs a global search range for each point, our goal is to find the adaptive search range of each point based on the local density information. In probability theory, the quantile function specifies the value of the random variable such that the probability of the variable being less than or equal to that value equals the given probability [18]. Thus, we exploit the quantile function, which specifies the distance from the center (mean) of the Gaussian distribution within a given probability, to find the adaptive search range of all points with the same quantile defined by $prob$. The quantile function F_i^{-1} under the Gaussian distribution centered on x_i in our proposed method is defined as

$$F_i^{-1} = \sigma_i \sqrt{2} erf^{-1}\left(2prob - 1\right), \quad (3)$$

where $prob \in (0.5, 1)$ is the probability, and $erf(\cdot)$ is the standard Gauss error function [19]. The quantile function F_i^{-1} is continuous and strictly monotonically increasing.

C. Mutual Nearest Neighbor

Further, we try to detect the points around the indistinguishable boundaries between clusters with varying densities. As shown in Fig. 2, there are two distinct clusters of different densities, and the blue dashed curve shows the boundary between dense and sparse regions, which is difficult to distinguish even by hand. We only highlight the representative points and mark the remaining points in gray. The adaptive search range of points in the dense region is generally small and vice versa. In the standard nearest neighbor search [20], since the *core point* x_i is within the adaptive search range of the *core point* x_j, x_j in the sparse region takes x_i in the dense region as the next search point. Then x_i and its neighbors are merged to the same cluster, though they are obviously in different clusters with varying densities.

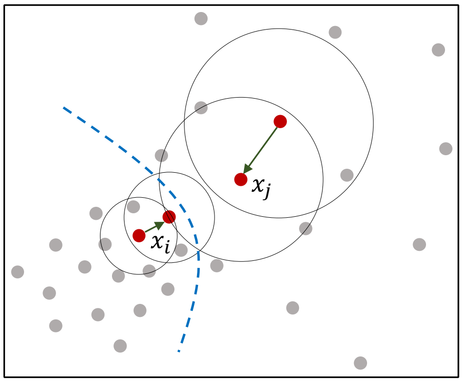


Figure 2. Assign points around the boundary.

To address this issue above, we propose a mutual nearest neighbor search as an effective tool to discover robust and reliable clusters. In Fig. 2, although x_i is in the adaptive search range of x_j, x_j is not in the adaptive search range of x_i. Therefore, We define the mutual nearest neighbor on the condition that both points are within each other's adaptive search range. Since we only need to consider pairs of points in each step rather than the entire data set, we can solely focus on the distance between these two points. The mutual nearest neighbor set MN_i of the point x_i is defined as

$$MN_i = \left\{x_j | dist\left(x_i, x_j\right) \leq \min(F_i^{-1}, F_j^{-1}), x_j \in D\right\}, \quad (4)$$

where x_j is a neighbor point of x_i and $dist\left(x_i, x_j\right)$ denotes the Euclidean distance between x_i and x_j.

D. DBADV Algorithm

For completeness of the paper, we first briefly review some notions of DBSCAN. The relationship between a *core point* and the neighbors within its global search range is called *directly density-reachable*. If any of these neighbors is a *core point* again, its neighbors are also transitively included in the same cluster. The relationship between all these neighbors and a series of these *core points* is called *density-reachable*. All points within the same cluster are called *density-connected*.

The proposed algorithm DBADV (cf.Algorithm 1) defines *clusters* as a region of smoothly varying density that is separated from other clusters by a remarkable change of the local density. DBADV can be generally divided into three parts: Firstly, we find the bandwidth as the local density information of each point through binary search with fixed perplexity (Line 3). The quantile function is then used to obtain the adaptive search range for each point (Line 4). Secondly, the notion of *directly density-reachable*

Algorithm 1: DBADV Algorithm

```
   Input: Data set $D = \{x_1, ..., x_n\}$; Perplexity $perp$;
          Minimum number of mutual nearest
          neighbors $MinPts$; Probability $prob$
   Output: Point labels $C = \{c_1, ..., c_n\}$
1  initialize each point label $c_i \in C$ as $undefined$,
    number of clusters $k = 0$;
2  foreach $x_i \in D$ do
3      $\sigma_i$ :=Binary search($D$,$perp$);
4      $F_i^{-1} := \sigma_i\sqrt{2}erf^{-1}(2prob - 1)$      // Eq.(3)
5  foreach $x_i \in D$ do
6      foreach $x_j \in D$ do
7          $dist(x_i, x_j) := \sqrt{\|x_i - x_j\|^2}$;
8          if $dist(x_i, x_j) \leq \min(F_i^{-1}, F_j^{-1})$ then
9              $MN_i := MN_i \cup \{x_j\}$         // Eq.(4)
10 foreach $x_i \in D$ do
11     if $c_i == undefined$ then
12         if $|MN_i| < MinPts$ then
13             $c_i := outlier$;
14         else
15             $k := k + 1$;
16             $c_i := k$;
17             $S_k := MN_i \setminus \{x_i\}$
18             foreach $x_q \in S_k$ do
19                 if $c_q == outlier$ then
20                     $c_q := k$;
21                 if $c_q == undefined$ then
22                     $c_q := k$;
23                     if $|MN_q| \geq MinPts$ then
24                         $S_k := S_k \cup MN_q$;
```

is redefined based on the quantile defined by $prob$ of each point to discover mutual nearest neighbors (Line 5-9), and the $MinPts$ is reconsidered as the minimum number of mutual nearest neighbors. Moreover, the notions of *density-reachable* and *density-connected* also change according to $MinPts$ and *directly density-reachable*. Finally, clustering is performed the same as DBSCAN (Line 10-24). The pseudo-code of the overall DBADV algorithm is described in Algorithm 1.

E. Computational Complexity Analysis

The asymptotic computational complexity of DBADV is composed of three main parts as follows. The first part is to perform a binary search on the bandwidth and obtain the adaptive search range for each object with a complexity of $O(n^2)$, where n is the number of objects; the second part is to find the mutual nearest neighbor for each object with a complexity of $O(n^2)$; finally, the complexity of discovering clusters is $O(n^2)$. Therefore, the overall asymptotic computational complexity of the proposed algorithm is $O(n^2)$, the same as DBSCAN.

IV. Experiments

We evaluate the performance of DBADV against a variety of state-of-the-art methods. Due to space limitations, we only can show a selection of the results. Source code, synthetic data sets, and supplementary material are available at https://dmm.dbs.ifi.lmu.de/cms/downloads.

A. Experimental Setup

We conduct experiments on two synthetic data sets and six real-world data sets. For baselines, we search for the best achievable clustering result in a reasonable range of parameters, or the default value of parameters recommended by the authors. For DBADV, extensive experiments have shown that fixing $prob$ to 0.977 and searching for $perp$ and $MinPts$ in the range of $[1, 30]$ is recommended. The evaluation metrics are performed through external measures Normalized Mutual Information (NMI) [24] and F-measure since class labels are available for the selected data sets. NMI and F-measure range in $[0, 1]$, where 1 and 0 denote perfect and arbitrary results, respectively. The Adjusted Normalized Mutual Information, a corrected-for-chance metric, gives similar scores to plain NMI (omitted due to space limitations). The supplementary material provides more detail about the experimental setup.

B. Synthetic Data

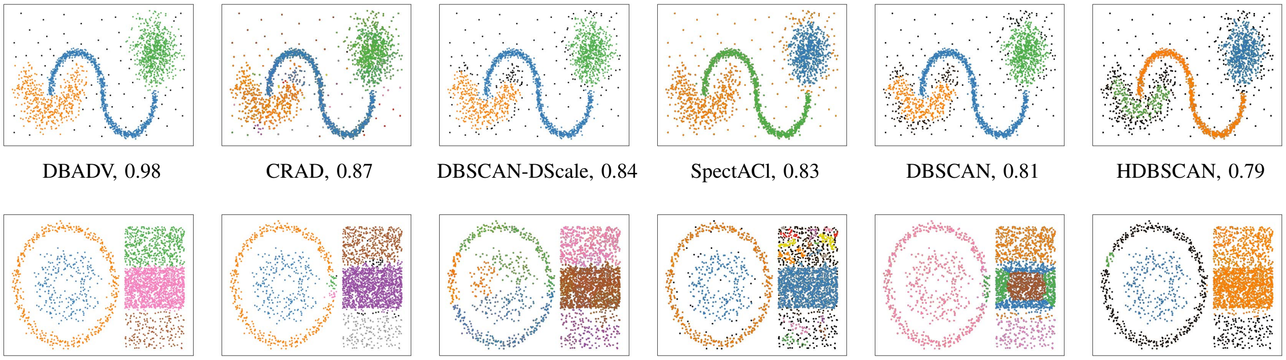
We generate synthetic data set *Shape3* with different densities, shapes, and sizes, which contains one lower half-moon with low density, one S-curve with high density, and one blob with low density that are all polluted by Gaussian noise. Besides, to explore the performance of clustering algorithms in the existence of outliers, we add 3% distinct global outliers evenly distributed on the *Shape3* and avoid the existing clusters. As shown in the top row of Fig. 3, DBADV is the only method that can accurately detect three clusters and almost all outliers. In addition, it pulls away from the runner-up by a large margin due to its robustness to noise and outliers. CRAD, as the runner-up in terms of NMI, can identify most of the major clusters. However, for the boundary points, especially between the high-density S-curve and the low-density lower half-moon, CRAD assigns them to the opposite cluster, while for outliers that fail to be identified, CRAD assigns them to separate smaller clusters. DBSCAN-DScale, DBSCAN, and HDBSCAN can accurately identify three clusters and outliers, but show weaknesses when dealing with boundary points of clusters with Gaussian noise. SpectACl also identifies three clusters but cannot find outliers. Instead, it allocates most outliers and the boundary points of the blob to the same low-density cluster.

In order to verify that DBADV can handle clustering in more complex environments, such as indistinguishable

Table I
NMI AND F-MEASURE ON REAL-WORLD DATA SETS (NA: NOT APPLICABLE). BEST VALUE IS IN BOLD; RUNNER-UP IS UNDERLINED.

Method	Seeds [21]		Dermatology [21]		Image Segment [21]		Page Blocks [21]		Crowdsourced [21]		warpPIE10P[1]	
	NMI	F-measure	NMI	F-measure	NMI	F-measure	NMI	F-measure	NMI	F-measure	NMI	F-measure
DBADV	**0.68**	**0.91**	**0.81**	**0.74**	**0.66**	**0.64**	**0.36**	**0.44**	**0.45**	**0.56**	**0.78**	**0.64**
DBSCAN [1]	0.60	0.79	0.62	0.52	0.63	0.59	0.24	0.26	0.38	0.23	0.60	0.09
OPTICS [3]	0.65	0.58	0.77	0.58	0.62	0.39	0.22	0.17	0.18	0.05	0.68	0.30
HDBSCAN [6]	0.41	0.48	0.64	0.33	0.65	0.57	0.19	0.29	0.33	0.16	0.69	0.32
CRAD [5]	0.42	0.65	0.46	0.33	NA	NA	0.10	0.29	NA	NA	0.41	0.06
DBSCAN-DScale [4]	0.65	0.84	0.80	0.59	0.65	0.61	0.27	0.32	0.38	0.23	0.68	0.40
SpectACl [7]	0.66	0.89	0.49	0.48	0.59	0.58	0.20	0.41	0.41	0.48	NA	NA
k-means [22]	0.67	0.89	0.79	0.55	0.62	0.62	0.13	0.26	0.26	0.32	0.30	0.30
Spectral [8]	0.02	0.22	0.03	0.11	0.40	0.38	0.14	0.31	0.18	0.30	0.08	0.03
Self-tuning [15]	0.38	0.17	0.46	0.26	0.00	0.04	NA	NA	0.32	0.27	0.62	0.29
Affinity Prop. [23]	0.51	0.50	0.67	0.59	0.59	0.33	0.13	0.36	0.23	0.26	0.64	0.49
Sync [11]	0.61	0.54	0.74	0.41	NA	NA	0.19	0.33	0.28	0.39	0.02	0.04

[1] https://jundongl.github.io/scikit-feature/datasets.html



DBADV, 0.98 CRAD, 0.87 DBSCAN-DScale, 0.84 SpectACl, 0.83 DBSCAN, 0.81 HDBSCAN, 0.79

DBADV, 0.97 DBSCAN-DScale, 0.91 Sync, 0.79 HDBSCAN, 0.76 SpectACl, 0.73 OPTICS, 0.70

Figure 3. Clustering performance of DBADV and top-5 baselines on synthetic data sets *Shape3* (top) and *Shape5* (bottom). The first term is the clustering algorithm, and the second term is the corresponding NMI, sorted by NMI.

boundaries and different data distribution, we generate synthetic data set *Shape5*. It contains two concentric circles polluted by Gaussian noise and three homogeneous rectangles with different densities and sizes. As shown in the bottom row of Fig. 3, DBADV is the only clustering algorithm that can accurately detect these five clusters and clearly distinguish the boundaries between clusters. DBSCAN-DScale is slightly inferior to DBADV, which can also recognize most regions of the clusters but fail to distinguish the boundary points, that is, between concentric circles and rectangles and between rectangles of different densities. For complex environments, many clustering algorithms show vulnerability. Sync splits two concentric circles of different densities into three clusters. HDBSCAN treats points in low-density as outliers or many small clusters. SpectACl also faces failure in handling boundary points and is prone to combine points in low-density into one cluster, such as two concentric circles, or to divide points in high-density into multiple clusters. OPTICS performs poorly, treating rectangles of different densities as one cluster and considering all points of low density as outliers.

C. Real-World Data

We select six real-world data sets from different areas of varying dimensionality ($d = 7 - 2420$), which are challenging due to their multivariate and high dimensionality. As shown in Table I, DBADV outperforms the other eleven baselines in terms of NMI and F-measure for all benchmarks, especially on warpPIE10P, which is a substantial margin compared to the runner up. In addition, DBADV also performs superior on high-dimensional data sets (e.g., warpPIE10P with 2420 dimensions) and large data sets (e.g., Crowdsourced with 10545 objects). With the extensive search range of three parameters, the overall performance of DBSCAN-DScale is closest to that of the DBADV. SpectACl and k-means follow closely in performance, earning runner-up on some data sets. DBSCAN, HDBSCAN, and Affinity Propagation have similar performances, showing moderate levels on all data sets. The performance of OPTICS and Sync varies greatly depending on the properties of data sets, which means that they perform well on some data sets and poorly on others. Spectral clustering, Self-tuning spectral clustering, and CRAD perform extremely poorly on some data sets, such as warpPIE10P. Some clustering algorithms cannot be applied in specific data sets due to limitations of their algorithm design, such as CRAD, Self-tuning spectral clustering, and Sync. SpectACl fails to process warpPIE10P because its parameter *eps* requires a larger search range.

V. Conclusion

We propose a novel Density-Based clustering algorithm for Adaptive Density Variation (DBADV), which can handle clusters of arbitrary shapes and sizes with varying densities, is robust to noise and outliers. It not only defines the local density information and thus finds the adaptive search range, but also designs a new metric to find the mutual nearest neighbor of each object for better detect objects around boundaries between clusters. Extensive experiments on challenging synthetic and real-world data sets have demonstrated that the proposed algorithm is effective and outperforms the other state-of-the-art clustering algorithms.

Acknowledgment

This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibilities for its content.

References

[1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *KDD*, 1996, pp. 226–231.

[2] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 1, no. 3, pp. 231–240, 2011.

[3] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure," in *SIGMOD*, 1999, pp. 49–60.

[4] Y. Zhu, K. M. Ting, and M. Angelova, "A Distance Scaling Method to Improve Density-based Clustering," in *PAKDD*, 2018, pp. 389–400.

[5] X. Huang and Y. R. Gel, "CRAD: Clustering with Robust Autocuts and Depth," in *ICDM*, 2017, pp. 925–930.

[6] R. J. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection," *ACM Trans. Knowl. Discov. Data*, vol. 10, no. 1, pp. 1–51, 2015.

[7] S. Hess, W. Duivesteijn, P. Honysz, and K. Morik, "The SpectACl of Nonconvex Clustering: A Spectral Approach to Density-Based Clustering," in *AAAI*, 2019, pp. 3788–3795.

[8] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an algorithm," in *NIPS*, 2001, pp. 849–856.

[9] S. Maurus and C. Plant, "Skinny-dip: Clustering in a Sea of Noise," in *KDD*, 2016, pp. 1055–1064.

[10] B. Schelling and C. Plant, "DipTransformation: Enhancing the Structure of a Dataset and Thereby Improving Clustering," in *ICDM*, 2018, pp. 407–416.

[11] J. Shao, X. He, C. Böhm, Q. Yang, and C. Plant, "Synchronization-Inspired Partitioning and Hierarchical Clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 893–905, 2013.

[12] J. Shao, C. Plant, Q. Yang, and C. Böhm, "Detection of Arbitrarily Oriented Synchronized Clusters in High-dimensional Data," in *ICDM*, 2011, pp. 607–616.

[13] C. Böhm and C. Plant, "Clustering by Synchronization," in *KDD*, 2010, pp. 583–592.

[14] W. Ye, S. Goebl, C. Plant, and C. Böhm, "FUSE: Full Spectral Clustering," in *KDD*, 2016, pp. 1985–1994.

[15] L. Zelnik-Manor and P. Perona, "Self-Tuning Spectral Clustering Lihi," in *NIPS*, 2004, pp. 1601–1608.

[16] R. Sennrich, "Perplexity Minimization for Translation Model Domain Adaptation in Statistical Machine Translation," in *EACL*, 2012, pp. 539–549.

[17] L. Van der Maaten and G. Hinton, "Visualizing Data using t-SNE Laurens," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[18] G. Steinbrecher and W. T. Shaw, "Quantile Mechanics," *Eur. J. Appl. Math.*, vol. 19, no. 2, pp. 87–112, 2008.

[19] A. Soranzo and E. Epure, "Very Simply Explicitly Invertible Approximations of Normal Cumulative and Normal Quantile Function," *Appl. Math. Sci.*, vol. 8, no. 87, pp. 4323–4341, 2014.

[20] D. A. Adeniyi, Z. Wei, and Y. Yongquan, "Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method," *Appl. Comput. Informatics*, vol. 12, no. 1, pp. 90–108, 2016.

[21] D. Dua and C. Graff, "{UCI} Machine Learning Repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[22] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[23] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[24] N. X. Vinh, J. Epps, and J. Bailey, "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, 2010.