Triangulation-based Spatial Clustering for Adjacent Data with Heterogeneous Density

Sihan Zhou[1], Daniel Vasiliu[2], and Guannan Wang[1]

[1]Department of Mathematics, William & Mary, Williamsburg, VA, 23187, U.S.A
[2]Data Science Program, William & Mary, Williamsburg, VA 23187, U.S.A.

Correspondence
Guannan Wang, William & Mary, Williamsburg, VA, 23187, U.S.A
Email: gwang01@wm.edu

Funding information
National Institutes of Health, Grant Number: 1R01AG085616-01
Simons Foundation, Grant Number: 963447

Acknowledgement
We gratefully acknowledge partial support for Guannan Wang's research from the National Institutes of Health under grant 1R01AG085616-01 and the Simons Foundation Mathematics and Physical Sciences-Collaboration Grant for Mathematicians (Grant No. 963447).

ABSTRACT

In numerous fields, including geography, bioinformatics, and software engineering, data often presents complex, nonlinear relationships and structures. Traditional clustering algorithms, which excel at identifying regularly shaped clusters such as ellipsoidal or spherical ones, falter when confronted with data comprising irregularly shaped groups, varying densities, and noise-perturbed inter-cluster connections. To address these challenges, we propose a novel graph-based Density and Triangulation-based Clustering (DTC) framework, articulated in three steps: (1) *density-based separation*, utilizing kernel density estimation to form preliminary clusters; (2) *Delaunay triangulation-based spatial clustering* for managing the data's nonlinear geometries and the issue of cluster adjacency; and (3) *noise management* via proximity to the nearest neighbors. The DTC framework's efficacy and practicality have been corroborated by experiments on both synthetic shape data and real-world datasets. Our numerical studies affirm that DTC is capable of discerning nested and contiguous clusters of heterogeneous densities with remarkable precision, yielding insights of substantive relevance.

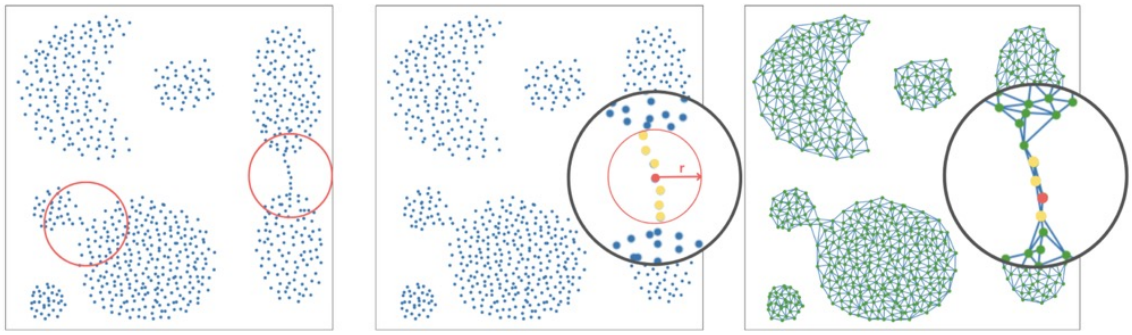
Keywords: Delaunay triangulation; KDE; DBSCAN; KNN; adjacent boundary

1 Introduction

Cluster analysis [1], an unsupervised learning technique, clusters data points by identifying inherent patterns without requiring pre-labeled training data. This approach is invaluable for uncovering latent structures across various fields, such as geography [2], bio-informatics [3], software engineering [4], despite its complexity due to the lack of explicit group information. Within this domain, spatial clustering is a niche that deals with geospatial, remote sensing, and image data analysis. The intricacy of spatial data, characterized by complex structures, non-uniform shapes, and diverse densities, presents significant challenges for conventional clustering methods to detect non-spherical cluster shapes.

To address the challenges faced by traditional partition-based clustering approaches such as K-means [12], Density-Based Clustering Algorithms (DBCLAs) have been introduced and gaining popularity for their ability to discern clusters of arbitrary shapes and densities. These algorithms employ a principle that estimates the density of a region and discerns clusters within the data space based on high-density areas. For instance, the Density-based Spatial Clustering of Applications with Noise (DBSCAN) [6], a prominent DBCLA, calculates the density of data points within a certain radius to determine a cluster. Building on DBSCAN's foundation, Hierarchical DBSCAN (HDBSCAN) [11] utilizes a hierarchical method that identifies clusters by merging nearby smaller clusters, thereby recognizing clusters and subclusters of varying densities and shapes. Similarly, the Ordering Points to Identify the Clustering Structure (OPTICS) algorithm [8] also leverages the comparison between the number of points lying within a certain reachability distance and the defined minimum cluster size to discern clusters and outliers. OPTICS further improved the parameter sensitivity problem by employing an augmented ordering of the dataset. On the other hand, several DBCLAs use probabilistic-based density computation to evaluate point density more accurately. The point-based DBCLAs, which include variants such as Statistical Information Grid in Data Mining (STING) [9] and Interactive Projected Clustering algorithm (IPCLUS) [10], employ probabilistic models and kernel density estimation to refine density calculation, thus increasing the precision in identifying irregular shapes and densities in data.

However, these methods sometimes struggle with closely spaced clusters, leading to the so-called "touching issue", as illustrated in Figure 1 (a). Recent research [13, 15] has incorporated Delaunay triangulation into clustering, using triangulation's max-min property [16] to define proximity relationships within data points better. This method effectively separates clusters connected by narrow links, utilizing the "removal of global effect" concept [13] to eliminate triangles that significantly deviate from the desired shape. In Figure 1 (b) and (c), we can see when two groups are connected by a "bridge", the triangulation approach can successfully separate them.



(a) Touching issue example (adjacent and bridge problem)

(b) Case example when DBSCAN fail to separate the groups with bridging problem

(c) Triangulation after removal of global effect which successfully solved the bridging problem

Figure 1: Illustration of the touching issue and the effectiveness of triangulation in solving this issue

Nevertheless, these approaches can be inadequate for data with highly irregular densities or nested clusters of varying densities. To overcome such limitations and generalize the clustering algorithm to accommodate all shapes and density relations within the data, we propose the Density and Triangulation-based Clustering (DTC) algorithm. This novel algorithm combines density estimation, Delaunay triangulation, and DBSCAN to balance accuracy and generalizability in clustering.

The remainder of this paper is organized to elaborate on the foundational concepts and tools employed in our proposed methodology. In Section 2, we give an overview of the essential concepts and tools used in the proposed method, including the KDE, Delaunay triangulation, a variant version of DBSCAN, and KNN for noise handling. Then, we present the development of our new clustering algorithm based on these toolkits. Section 3 presents simulation results comparing our proposed method with several popular algorithms in the literature. This section also includes an actual data application on the clustering analysis of $PM_{2.5}$ air pollution in the United States to demonstrate our algorithm's efficacy in real-life scenarios. The final section reviews the distinctive aspects of the DTC and discusses prospective enhancements.

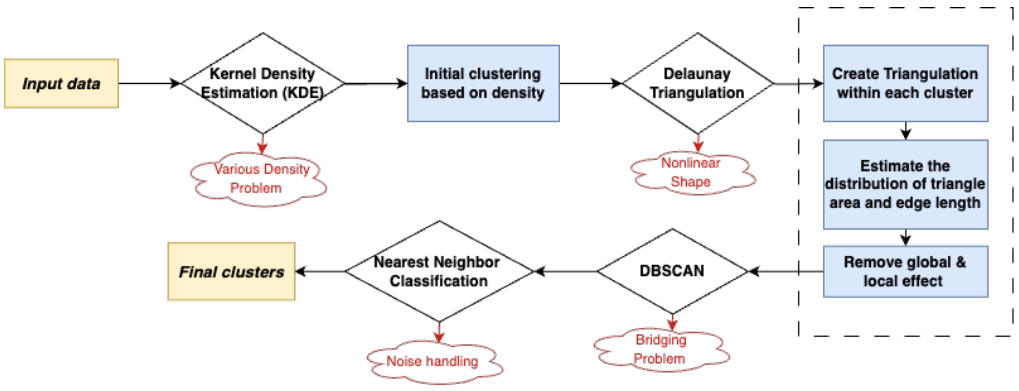


Figure 2: Flowchart of the Density and Triangulation based Clustering Algorithm.

2 Methodology and Implementation Details

Figure 2 illustrates the entire flowchart of the proposed DTC algorithm. The initial phase of the proposed method involves applying Kernel Density Estimation (KDE) to the dataset. Subsequently, a density-based partitioning algorithm, as delineated in Algorithm 1, is employed to preliminarily separate the data into distinct clusters. This initial separation is primarily based on the varying density levels observed across the dataset.

Following the initial clustering, the process entails constructing Delaunay triangulations within each identified cluster. We mitigate the "global effect" [13] by systematically excluding triangles that exhibit disproportionately large edge lengths and area sizes. This exclusion criterion is determined based on the statistical distribution of both edge lengths and area sizes of all the triangles. Then, we remove all the outlying edges based on the distribution of the edges connecting to each vertex. Such an approach is instrumental in addressing the "local effect" and is pivotal in facilitating the delineation of clusters with non-linear geometries.

Building upon the refined set of triangles, the methodology incorporates a modified version of DBSCAN algorithm. Contrary to the conventional DBSCAN approach that relies on a predefined radius (epsilon value) for cluster formation, this variation utilizes the count of neighboring vertices connected to each point, as determined by the final triangulation. This integration of Delaunay triangulation with DBSCAN enhances the algorithm's capability to effectively navigate the complexities associated with the bridging phenomena in data clustering.

In the final stage of the process, data points identified as "noise" or outliers in the DBSCAN phase are reassigned labels. This reassignment is based on the proximity of these points to their nearest labeled neighbors, thereby ensuring a more accurate classification of boundary points and enhancing the overall robustness of the clustering algorithm.

2.1 Density-based Separation

Density estimation

Density estimation is a critical process in determining the probability distribution of a random variable based on a collection of data points. Its significance is particularly pronounced in resolving the issue of "cluster touching" in datasets with varying densities. In such scenarios, clusters with different densities may be erroneously perceived as singular entities without the intricate separation afforded by density estimation. This technique becomes increasingly vital as datasets evolve in complexity, exhibiting irregular and nested structures. Density estimation facilitates the nuanced separation of adjacent clusters, a task that may not be fully achievable through triangulation-based methods alone. By incorporating density estimation, the DTC algorithm attains a level of generalizability, effectively capturing relationships between data points irrespective of the variations in shape or density present within the dataset. KDE [17], a widely utilized method in density estimation, finds applications in diverse areas such as image processing, finance, anomaly detection, and environmental science. In our proposed algorithm, the Gaussian kernel is employed to estimate the density of each data point.

Separation algorithm

Density estimation is a critical process in determining the probability distribution of a random variable based on a collection of data points. Its significance is particularly pronounced in resolving the issue of "cluster touching" in datasets with varying densities. In such scenarios, clusters with different densities may

be erroneously perceived as singular entities without the intricate separation afforded by density estimation. This technique becomes increasingly vital as datasets evolve in complexity, exhibiting irregular and nested structures. Density estimation facilitates the nuanced separation of adjacent clusters, a task that may not be fully achievable through triangulation-based methods alone. By incorporating density estimation, the DTC algorithm attains a level of generalizability, effectively capturing relationships between data points irrespective of the variations in shape or density present within the dataset. KDE [17], a widely utilized method in density estimation, finds applications in diverse areas such as image processing, finance, anomaly detection, and environmental science. In our proposed algorithm, the Gaussian kernel is employed to estimate the density of each data point.

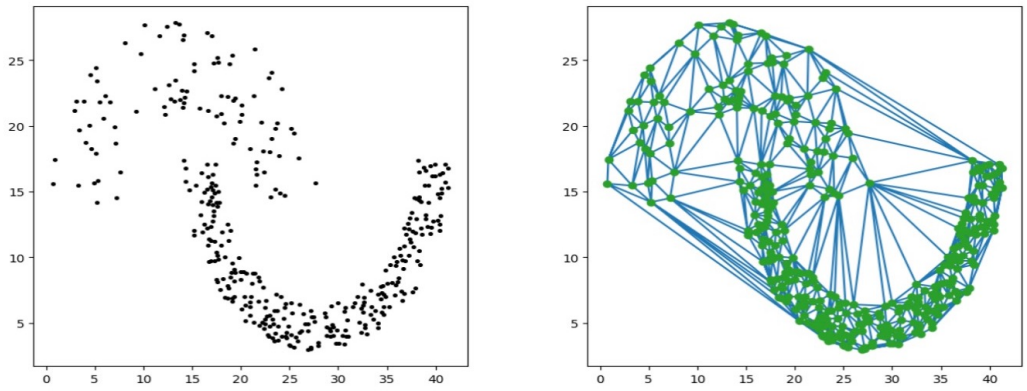
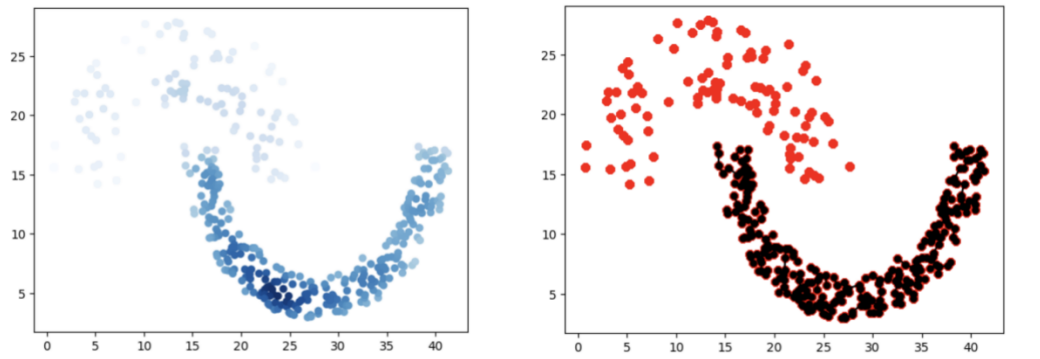


Figure 3: An example of data set with various density

Let $\mathcal{P}$ be all the data points, $\mathcal{C}_i$ denotes the set of points that are put into the same cluster i, and $\mathcal{U}$ denotes the remaining points. Initially, we set $\mathcal{C} = \emptyset$, $\mathcal{U} = \mathcal{P}$. We first estimate the density for each data point. Among all the data points, we find point $p \in \mathcal{U}$ with the highest density and put it in $\mathcal{C}$. Then we find another point $p' \in \mathcal{U}$ that is closest to p, and move p' into the set C, record the corresponding step size s. Then, within $\mathcal{U}$, we look for the next point closest to any point in S but only move it into the cluster if the new step size is not significantly different compared to the distribution of all the past step sizes. We repeat the process until $\mathcal{U} = \emptyset$.

Considering Figure 4 (a), it becomes evident that the two moon-shaped data groups warrant classification into distinct clusters, primarily owing to the discernible differences in their density profiles. The implementation of the aforementioned algorithm facilitates this distinction, enabling the identification of clusters that exhibit varying density characteristics. This capability surpasses what could be achieved through methods that exclusively depend on local proximity relationships, highlighting the efficacy of the algorithm in distinguishing between clusters with heterogeneous density distributions.



(a) kernel density estimation on each data point

(b) the initial clusters obtained after the density-based separation

Figure 4: Perform density-based partitioning on the same data set

Algorithm 1 Desnity-based separation

Input: data points P with their corresponding kernel density D
Initialization: $\mathcal{C}_i = \emptyset$, $i = 0$, $\mathcal{U} = P$, $\mathcal{L} = \emptyset$
while $\mathcal{U} \neq \emptyset$ **do**
    Find the point $p \in \mathcal{U}$ with the maximum kernel density in D and the closest point $p' \in \mathcal{U}$.
    Move p, p' from $\mathcal{U}$ to $\mathcal{C}$. Record the step size between p and p' as l in $\mathcal{L}$.
    Find the next closest point $p' \in \mathcal{U}$ to the set C and calculate the step size l
    if $l > (\mu(\mathcal{L}) + c\sigma(\mathcal{L}))$ or $l > Q_3(\mathcal{L}) + c'IQR(\mathcal{L})$ **then**
        $i+ = 1$
    else
        Move p' from $\mathcal{U}$ to $\mathcal{C}$. Record the corresponding l into $\mathcal{L}$.
    end if
end while

2.2 Delaunay triangulations-based DBSCAN

Delaunay triangulations

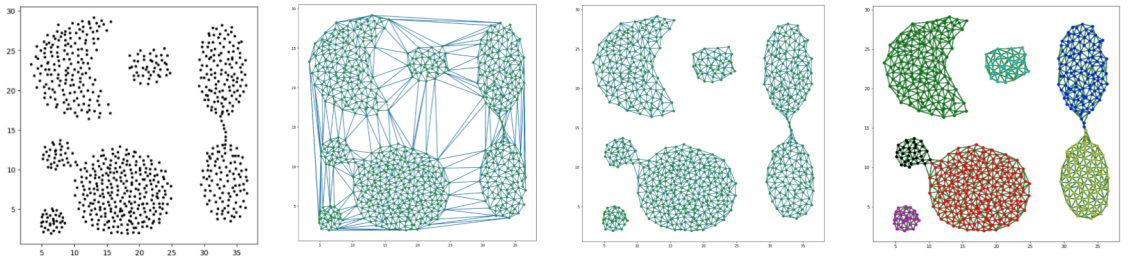
Given a set of points, Delaunay triangulation [14] constructs triangles in such a manner that no point lies within the circumcircle of any triangle. A key characteristic of Delaunay triangulations is their inherent ability to maximize the minimum angle among all the angles of the triangles in the triangulation. This property is instrumental in mitigating the formation of excessively narrow or 'skinny' triangles, except when such configurations are necessary. Delaunay triangulations can be generated using a variety of software packages, each tailored to specific computational environments. Notable examples include the `Triangulation` [20] package in `R`, the `Distmesh` program [19] for `Matlab`, and the `scipy.spatial.Delaunay` [21] package in `Python`. These tools offer robust solutions for implementing Delaunay triangulations in diverse research and application domains.

Removal of Global & Local Effect

Our methodology commences with a global estimation of the distribution of both the area and edge length across all triangulated structures. Leveraging the inherent attributes of triangulation, it becomes feasible

to exclude triangles that exhibit excessively long edges or disproportionately large areas. This exclusion is crucial for unveiling potential clusters within the data. In scenarios where data points within a cluster are relatively uniformly distributed, the resultant triangulation predominantly consists of triangles that closely approximate equilateral forms. Consequently, it becomes pertinent to eliminate edges that are outliers in the context of their connection to a single vertex. This process constitutes a comprehensive pre-processing step in our clustering algorithm, addressing both global and local anomalies.

Figure 5 (c) shows the resulting set of triangles after we remove the global and local effects, followed by the distribution of areas and edge length of triangles in (b). Now, we could perform DBSCAN by choosing minimum points, a value of points required to form a cluster, as the hyperparameter. See Algorithm 2 for a detailed procedure description.



(a) Original data with adjacent problem and touching issue

(b) Delaunay Triangulation on the data

(c) Set of triangles after removing global effect

(d) DBSCAN clustering result after KNN

Figure 5: Example of using Triangulation-based DBSCAN to solve adjacent problems

Algorithm 2 Triangulation-based DBSCAN

Input: data points X, a set of triangles T, and the hyperparamter $minPts$.
Initialization: Create a stack $\mathcal{C} = \emptyset$, a list of unvisited points $\mathcal{U} = X$. Set cluster index $i = 0$.
while $\mathcal{U} \neq \emptyset$ **do**
    Randomly choose a point from $\mathcal{U}$ and add it into $\mathcal{C}$.
    while $\mathcal{C} \neq \emptyset$ **do**
        Pop a point p from $\mathcal{C}$, add all the vertices connected to p into $\mathcal{C}$.
        Count the number vertices connected to p, compare it with $minPts$.
        if number of vertices $> minPts$ **then**
            mark p as a member of cluster i
        else
            mark p as a noise
        end if
    end while
    i += 1
end while

2.3 Relabeling of Noise

Up to this juncture in our analysis, we have achieved an approximately precise clustering outcome by integrating KDE, Delaunay triangulation, and DBSCAN. However, a notable limitation arises from the DBSCAN's hyperparameter, specifically "minPts" (minimum points), which tends to result in the erroneous

classification of data points near cluster boundaries as noise. To address this issue, we subsequently apply a K-Nearest Neighbors (KNN) classification algorithm. This additional step is instrumental in reassessing the data points previously designated as noise within the clustering framework, thereby refining the overall accuracy of the clustering results.

3 Numerical Experiments

3.1 Synthetic Examples

In this subsection, our focus is on evaluating the performance of the proposed DTC algorithm through its application to several benchmark synthetic datasets. We compare DTC's efficacy with that of established clustering algorithms, including K-Means, DBSCAN, HDBSCAN, and OPTICS. The datasets utilized for this comparison include the "Aggregation" dataset, "A.K. Jain's Toy problem" dataset, and "Zahn's Compound datasets" [22].

Figure 6 illustrates the "Aggregation" dataset, a classic example used to assess clustering algorithms' performance on complex spatial data. This dataset is characterized by its non-linear structure and the presence of both adjacent and bridging clusters. Traditional clustering algorithms like K-Means are unable to cluster non-linear shapes effectively, and other density-based methods struggle to segregate adjacent clusters. In our DTC methodology, as shown in Figure 5, we initially eliminate relatively long edges, which enables a more precise separation of touching clusters and bridges, especially after the global and local removal of extraneous triangles in preparation for DBSCAN.

For the subsequent datasets, which feature adjacent groups with varying densities, the DTC algorithm employs a density-based separation mechanism to divide the data into clusters initially using KDE, followed by triangulation-based DBSCAN. The kernel density for each data point is computed using the Python library *scipy.stats.gaussian_kde* with the bandwidth selected by Scott's Rule [18] to calculate the kernel density for each data point. The hyperparameter *min_sample* which specifies the minimum cluster size is set to 12 in these examples.

The "Jain's Toy Problem" dataset features two moon-shaped clusters in close proximity, posing challenges for traditional methods to both accurately assign the closely situated points to the correct cluster and maintain the integrity of each moon-shaped group as a distinct cluster due to their different densities. In the comparison shown by Figure 7, we can see that HDBSCAN and DBSCAN created too many clusters because they did not appropriately handle the density relationship between the two clusters. But with the aid of clustering based on kernel density estimation, DTC effectively solves the problem and achieves nearly perfect accuracy.

In Figure 8, the "Zahn's Compound" dataset contains not only adjacent but also nested non-linear clusters with varied densities. Density estimation is essential for segregating the nested groups, particularly in the right and lower left corners, while triangulation-based DBSCAN addresses the adjacency of the two non-linear-shaped groups. The comparative methods exhibit shortcomings in distinctly separating closely situated groups or differentiating the inner cluster from its surrounding counterpart. However, the DTC method adeptly overcomes these challenges by integrating a KDE-based clustering approach with triangulation-based DBSCAN.

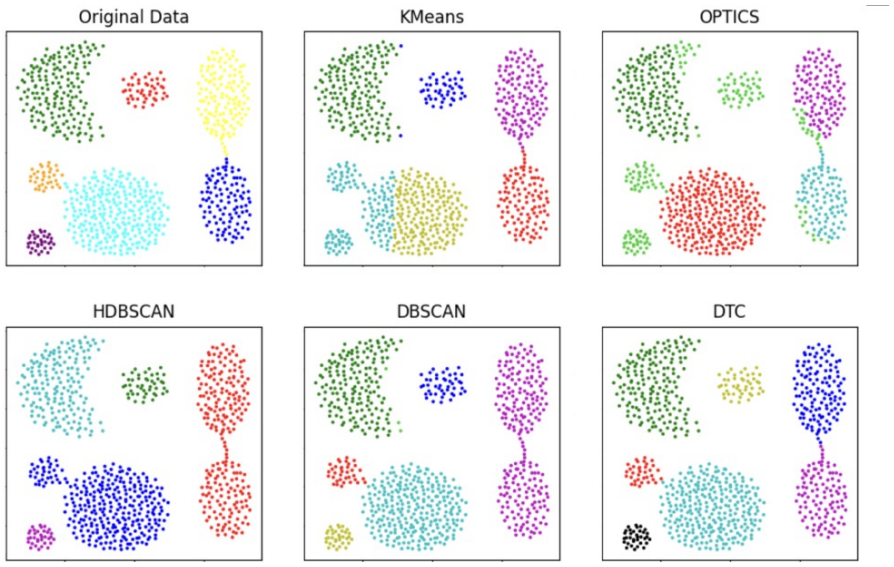


Figure 6: Comparison of the performance on the "Aggregation" data set

	Kernel K-mean	HDBSCAN	DBSCAN	OPTICS	DTC
Aggregation Dataset	86.29%	82.74%	86.42%	84.77%	99.62%
A.K. Jain's Toy Problem	78.55%	80.70%	65.15%	98.93%	100.00%
Zahn's Compound Dataset	57.89%	84.71%	80.95%	91.23%	99.50%

Table 1: Accuracy comparison table

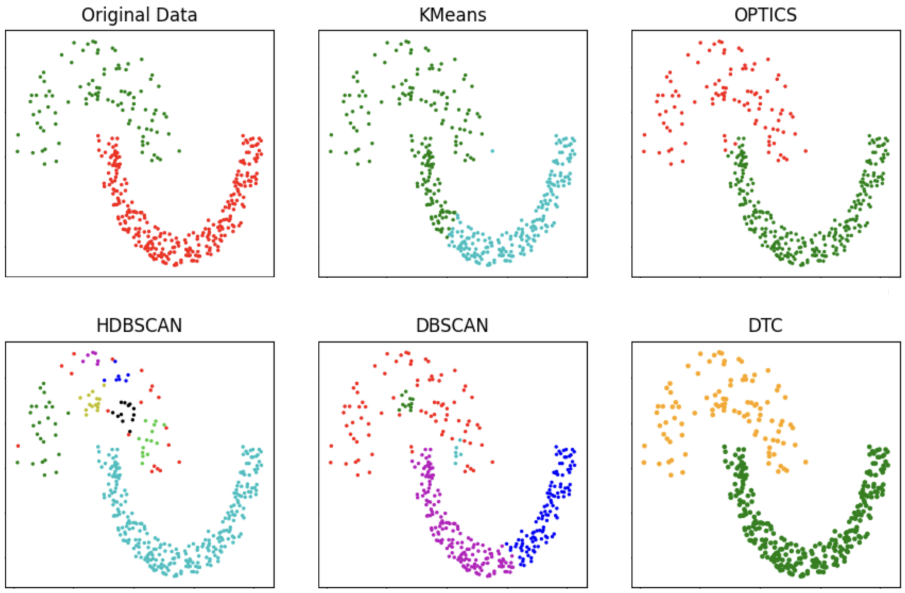


Figure 7: Comparison of the performance on the " A.K. Jain's Toy Problem" data set

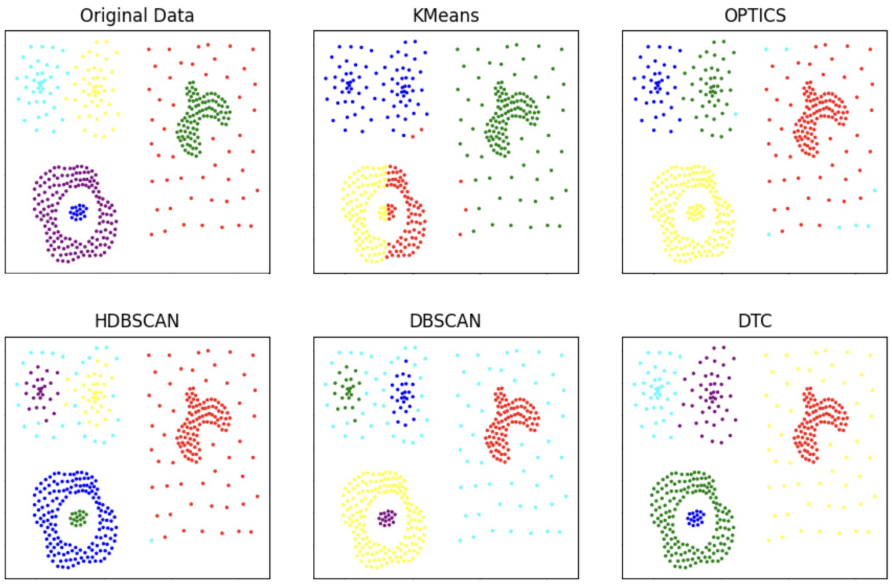


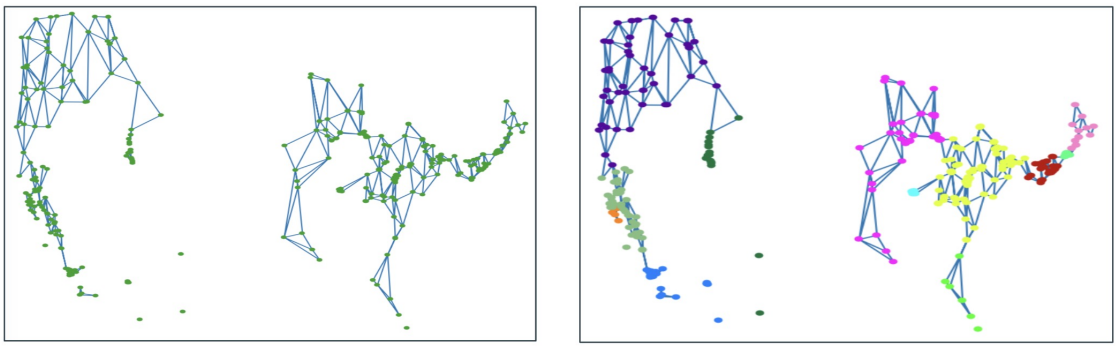
Figure 8: Comparison of the performance on "Zahn's Compound" data set

3.2 U.S. $PM_{2.5}$ Analysis

To further examine the applicability of the proposed DTC algorithm to real data, we applied the proposed algorithm to meteorological data. We used the daily mean surface concentrations of total $PM_{2.5}$ for the year 2011 obtained from the United States Environmental Protection Agency. Among all the observed sites, we selected the locations where the corresponding $PM_{2.5}$ value is greater than $12ug/m^3$, the threshold value given by the U.S. Environmental Protection Agency (EPA) to differentiate good and potentially harmful air quality.

Upon implementing our algorithm as depicted in Figure 9, the clustering results from Figure 10 demonstrate that the nonlinear-shaped $PM_{2.5}$ spatial data points are separated into 12 clusters, providing insightful information on the air quality across various regions in the U.S. The dark green, purple, cyan, and orange clusters specifically pinpoint major urban centers, including New York, Philadelphia, San Jose, and St Louis. Despite these cities being represented by closely situated data points that could be prone to misclassification, our algorithm effectively differentiates them.

Furthermore, Figure 11 showcases the varying pollution levels across each cluster. Cities on the West Coast, especially in California, display higher pollution levels compared to those in the midland and on the East Coast. Among eastern cities, areas encompassing Philadelphia, New York, and St Louis show elevated $PM_{2.5}$ values. This clustering result also aligns with the "Most Polluted City" rankings by the American Lung Association [23].



(a) PM2.5 data after global & local removal

(b) PM2.5 data's final clustering result

Figure 9: Apply the triangulation algorithm to PM2.5 data

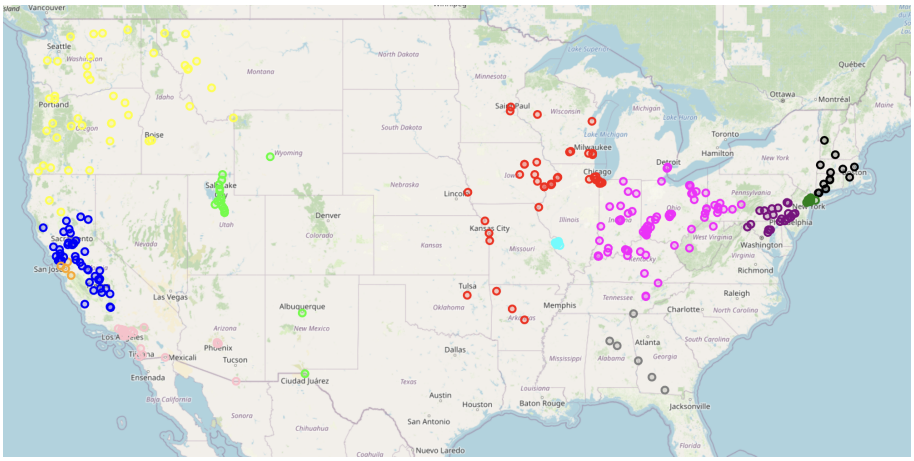


Figure 10: Clustering result on the 2011's PM2.5 data in the United States

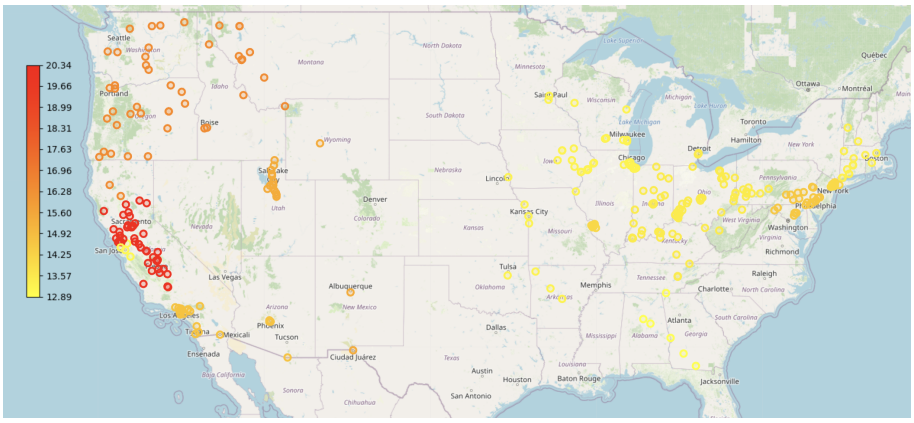


Figure 11: Average PM2.5 level for each clusters

4 Conclusion and Future Work

The foregoing discussions and visual demonstrations have underscored the key strengths of the proposed DTC algorithm. Its efficacy is particularly pronounced in processing data points that are in close proximity and exhibit complex structural and density characteristics. The DTC algorithm's exceptional ability to address the "touching" problem, a common challenge in spatial clustering, positions it as a highly competitive alternative to other existing clustering methodologies.

In the ever-evolving landscape of machine learning and pattern recognition, there is an escalating demand for advanced spatial clustering tools capable of efficiently handling datasets with intricate and non-uniform structures. Responding to this demand, our future endeavors include the development of Fast-DTC, an iteration of the DTC algorithm with enhancements in computational speed and memory efficiency. Fast-DTC aims to significantly elevate the performance in clustering large-scale and complex datasets. This advancement is anticipated to broaden the scope of DTC's applicability in various domains within machine learning, offering solutions to more challenging and diverse clustering scenarios. The implementation of Fast-DTC has the potential to be a pivotal contribution to the field, addressing the growing complexity and size of datasets encountered in contemporary machine learning applications.

References

[1] D. B. Ramey, Nonparametric clustering techniques, *Encyclopedia of Statistical Science*. 6 (1985), 318–319.

[2] W. Gesler, The uses of spatial analysis in medical geography: A review, *Social Science & Medicine*. 23 (1986), no. 10, 963-973.

[3] D. J. Miller, Y. Wang, and G. Kesidis, Emergent unsupervised clustering paradigms with potential application to bioinformatics, *Frontiers in Bioscience-Landmark*. 13 (2008), no. 2, 677-690.

[4] M. Shtern and V. Tzerpos, Clustering methodologies for software engineering, *Advances in Software Engineering*. 2012.

[5] T. H. Grubesic, R. Wei, and A. T. Murray, Spatial clustering overview and comparison: Accuracy, sensitivity, and computational expense, *Annals of the Association of American Geographers*. 104 (2014), no. 6, 1134-1156. https://doi.org/10.1080/00045608.2014.958389

[6] M. Ester, H. P. Kriegel, J. Sander, and X. Xiaowei, A density-based algorithm for discovering clusters in large spatial databases with noise, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. (1996), 226-231.

[7] R. J. G. B. Campello, D. Moulavi, and J. Sander, Density-Based Clustering Based on Hierarchical Density Estimates, *Advances in Knowledge Discovery and Data Mining*. 7819 (2013). ISBN: 978-3-642-37455-5.

[8] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, OPTICS: ordering points to identify the clustering structure, *SIGMOD Rec.* 28 (1999), no. 2, 49-60. https://doi.org/10.1145/304181.304187.

[9] W. Wang, J. Yang, and R. Muntz, Sting: a statistical information grid approach to spatial data mining, *Proceedings of the 23rd International Conference on Very Large Data Bases*. (1997), 186–195.

[10] C. C. Aggarwal, A human-computer interactive method for projected clustering, *IEEE Transactions on Knowledge and Data Engineering*. 16 (2004), no. 4, 448-460.

[11] R. J. Campello, D. Moulavi, and J. Sander, Density-based clustering based on hierarchical density estimates, *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference*. (2013), 160–172.

[12] P. Bhattacharjee and P. Mitra, A survey of density-based clustering algorithms, *Front. Comput. Sci.*. 15 (2021), 151308. https://doi.org/10.1007/s11704-019-9059-3

[13] J. Kim and J. Cho, Delaunay triangulation-based spatial clustering technique for enhanced adjacent boundary detection and segmentation of LiDAR 3D point clouds, *Sensors*. 19 (2019), no. 18, 3926. https://doi.org/10.3390/s19183926

[14] B. Delaunay, Sur la sphère vide, *Bull. Acad. Sci. USSR, Classe Sci. Math. Nat.*. 6 (1934), 793–800.

[15] Q. Liu, M. Deng, Y. Shi, and J. Wang, A density-based spatial clustering algorithm considering both spatial proximity and attribute similarity, *Computers & Geosciences*. 46 (2012), 296–309. 10.1016/j.cageo.2011.12.017.

[16] O. R. Musin, Properties of the Delaunay triangulation, Dept. of Cartography and Geoinformatics, Moscow State University. (n.d.).

[17] E. Parzen, On estimation of a probability density function and mode, *The Annals of Mathematical Statistics*. 33 (1962), no. 3, 1065-1076. https://doi.org/10.1214/aoms/1177704472.

[18] D.W. Scott, "Multivariate Density Estimation: Theory, Practice, and Visualization", John Wiley & Sons, New York, Chicester, 1992.

[19] P.-O. Persson and G. Strang, A Simple Mesh Generator in MATLAB, *SIAM Review*. 46 (2004), no. 2.

[20] G. Wang, L. Wang, and M. J. Wang, Triangulation for a 2D domain, R package version 1.0, (2022). Available at https://github.com/funstatpackages/Triangulation

[21] SciPy developers, *scipy.spatial.Delaunay*, SciPy Reference Guide. Version 1.8.0, (2023). Available at https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.Delaunay.html

[22] P. Fänti and S. Sieranoja, K-means properties on six clustering benchmark datasets, *Applied Intelligence*. 48 (2018), no. 12, 4743–4759. https://doi.org/10.1007/s10489-018-1238-7

[23] American Lung Association, Most polluted cities: State of the Air. https://www.lung.org/research/sota/city-rankings/most-polluted-cities (n.d.)