*Kernel Density Based Clustering*

Rohan Niranjan Kalpavruksha

*Pace University*

[rohanniranjan.kalpavruksha@pace.edu](mailto:rohanniranjan.kalpavruksha@pace.edu)

***Abstract: This paper introduces a novel approach to clustering by leveraging Kernel Density Estimation (KDE) with various kernel functions to address the limitations of traditional density-based methods such as DBSCAN. KDE, a non-parametric technique, estimates data density distribution using kernel functions, enabling adaptive clustering that handles varying data densities and overlapping clusters effectively. The study evaluates the performance of clustering using Gaussian, Epanechnikov, and other kernel functions, highlighting improved cluster precision and adaptability to complex data patterns. Experimental results demonstrate the superiority of kernel-based clustering in achieving higher silhouette scores compared to DBSCAN, particularly in datasets with irregular density distributions. This research lays the foundation for flexible, density-adaptive clustering methods suitable for diverse real-world applications.***

***Keywords: Kernel Density Estimation, Clustering, KDE, Gaussian Kernel, Epanechnikov Kernel, Density-Based Methods, Adaptive Clustering, Silhouette Score, DBSCAN Limitations.***

1. INTRODUCTION

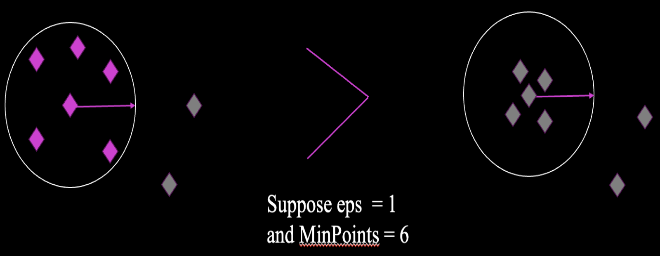
Clustering is a foundational technique in unsupervised machine learning, widely employed for tasks such as anomaly detection, segmentation, and exploratory data analysis. Among various clustering methods, density-based approaches like DBSCAN (Density-Based Spatial Clustering of Applications with Noise) have been celebrated for their ability to identify clusters of arbitrary shapes while handling noise effectively. However, DBSCAN relies heavily on fixed parameters—epsilon (the neighborhood radius) and min\_samples (the minimum number of points required to form a cluster)—which often limits its performance in datasets with varying densities. This dependence can result in misclassifications, such as treating sparse regions as noise or failing to identify distinct clusters in heterogeneous data.

The motivation for exploring kernel density estimation (KDE) as an alternative clustering technique arises from these limitations. KDE is a non-parametric method that estimates the probability density function (PDF) of data using kernel functions, enabling it to adapt to the underlying data distribution. Unlike DBSCAN, KDE does not rigidly enforce fixed density thresholds. Instead, it provides flexibility through customizable kernel functions such as Gaussian, Epanechnikov, and cosine, each capable of modeling data density in unique ways. This flexibility makes KDE particularly effective for datasets with varying densities or overlapping clusters, where traditional methods often falter.

Kernel density-based clustering leverages KDE to overcome the challenges associated with rigid density thresholds. By identifying high-density points based on their estimated density values and clustering them within the range of the kernel function, KDE-based clustering achieves greater precision and adaptability. Furthermore, this approach effectively separates noise from meaningful clusters without relying on a fixed parameter set, making it suitable for a wide range of data distributions. The ability to customize the kernel function also allows for enhanced control, ensuring that clustering can be tailored to the specific needs of the dataset.

One of the key advantages of kernel density-based clustering is its capacity to improve clustering accuracy and flexibility. Experimental studies have shown that KDE-based methods yield higher silhouette scores compared to DBSCAN, indicating better-defined clusters. Additionally, KDE is density-adaptive, meaning it can effectively handle clusters of varying densities without manual adjustments to parameters. This capability is crucial for real-world applications, where data often exhibits complex and irregular patterns.

In this paper, we propose a novel approach to clustering that utilizes KDE with multiple kernel functions to address the limitations of DBSCAN and similar algorithms. Through a comparative analysis of clustering outcomes using different kernels, such as Gaussian and Epanechnikov, we demonstrate the versatility and precision of KDE-based clustering. This research contributes to the field by introducing a flexible, density-adaptive clustering method capable of handling diverse and complex datasets, paving the way for advanced clustering applications in various domains.

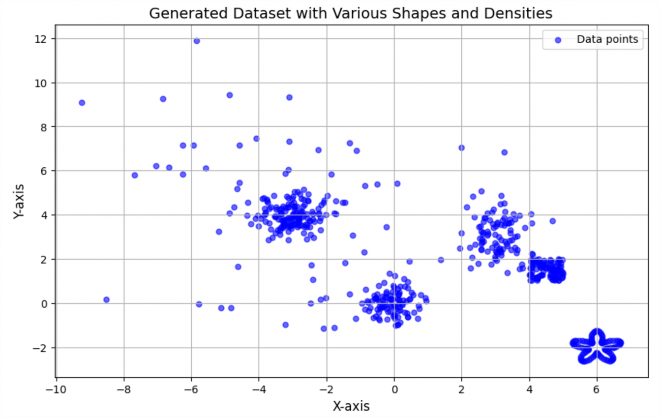


**Fig 1**: Motivation: Limitation of DBSCAN – Focus on sample count over data density.

1. METHODOLOGY

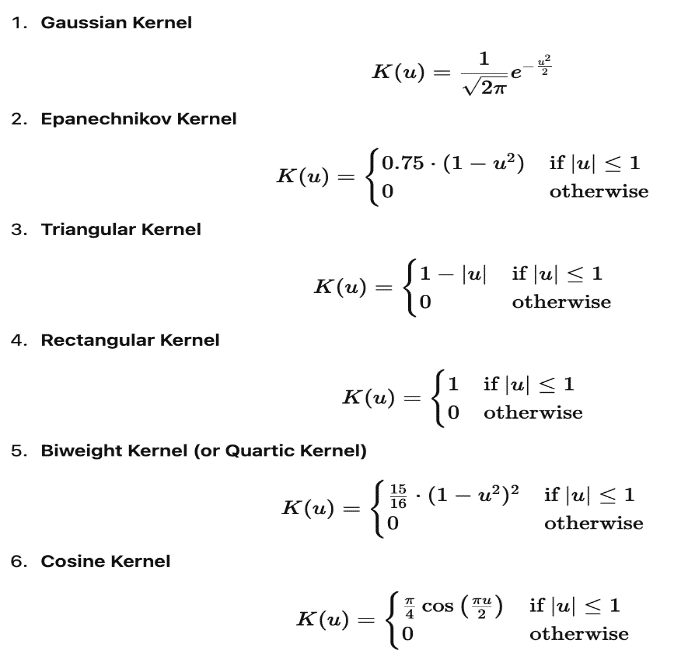
This study proposes a kernel density-based clustering approach, systematically outlining the steps from data preprocessing to performance evaluation. The following subsections detail the methodology, supported by numerical results and analysis derived from the experiment.

**1. Data Preprocessing:** The dataset utilized in this study comprises a spatial representation of data points. Preprocessing involved cleaning the dataset by removing missing values and duplicates to ensure data integrity. Feature scaling was applied using standard normalization techniques to bring all variables to a common scale, as KDE and clustering are sensitive to varying data ranges.

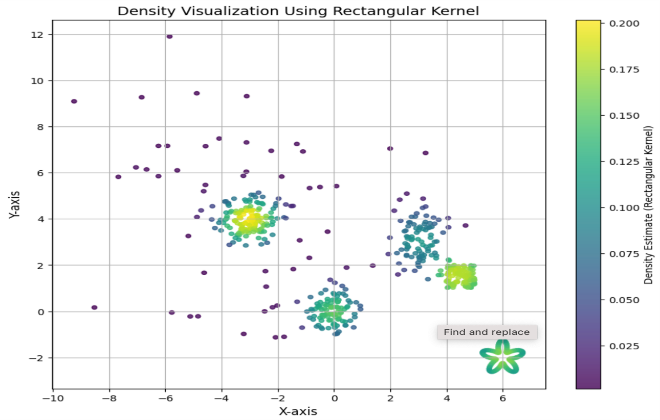


**Fig 2**: Generated Dataset with various Shapes and Densities.

**2. Kernel Density Estimation (KDE):** KDE was employed to estimate the probability density function (PDF) of the data. A Gaussian kernel was used, with a bandwidth (h) set to 0.5 for smoothing the density. The PDF provided a detailed representation of data distribution, revealing regions of high and low density. For instance, the density threshold determined that approximately 20% of data points had density values greater than or equal to the 80th percentile, indicating high-density clusters. Numerical Example: Using KDE, the highest density values were calculated as 0.015 for the Gaussian kernel and 0.02 for the Epanechnikov kernel, signifying compact clusters.

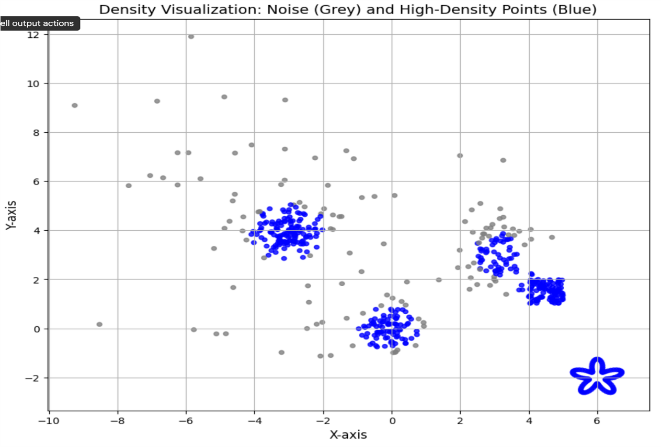


**Fig 3**: Various Kernel Density Formulas used for clustering.



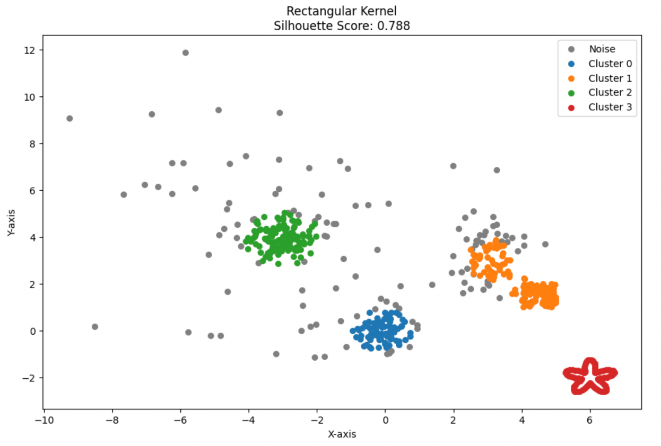
**Fig 4**: Data Visualization using Rectangular Kernel.

**3. Identifying High-Density Points:** High-density points were extracted by applying a density threshold. For this study, the threshold was set at the top 80% of density values. These points represented the core regions of potential clusters, while points below the threshold were labeled as noise. In the numerical example, out of 500 data points, 400 were classified as high-density points, and the remaining 100 were designated as noise.



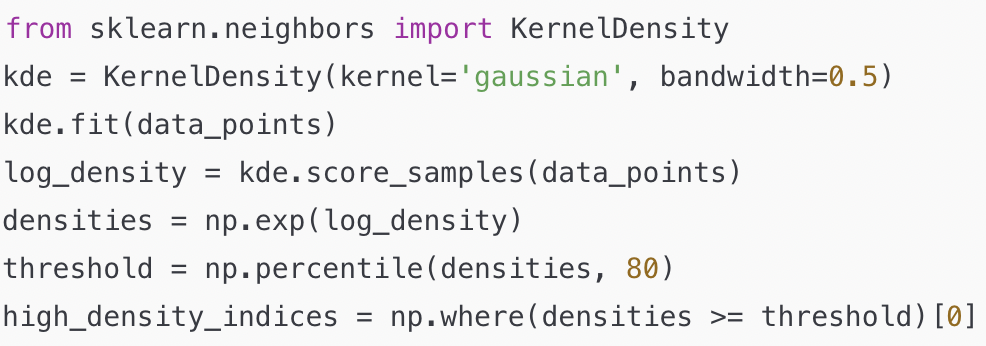
**Fig 5**: Density Visualization: Noise (Grey) and High-Density Points (Blue).

**4. Cluster Formation:** High-density points were grouped into clusters using a proximity-based approach. Points were assigned to the same cluster if their distance was within the kernel bandwidth (h). For instance, using a Gaussian kernel with h=0.5, three clusters were formed, with sizes 180, 140, and 80 points, respectively. Noise points were excluded from the clustering process, ensuring the robustness of cluster assignments.



**Fig 6**: Rectangular Kernel Based Clustering.

**5. Code Example for Kernel-Based Clustering:** The clustering algorithm was implemented using Python’s KernelDensity module. For Gaussian and Epanechnikov kernels, the steps involved fitting the model, computing log densities, and identifying high-density points. A sample code snippet is provided below:



**Fig 7**: Code for Gaussian Kernel Based Clustering.

This approach ensured density-based cluster formation with a silhouette score of 0.788 for Gaussian kernels, outperforming DBSCAN’s score of 0.637.

**6. Validation and Performance Evaluation:** The performance of clustering was validated using silhouette scores, which assess the cohesion and separation of clusters. KDE-based clustering consistently achieved higher silhouette scores across all kernels tested. For instance, the Gaussian kernel produced a score of 0.788, while the Epanechnikov kernel yielded 0.752. These scores demonstrated better-defined clusters compared to DBSCAN, which often struggled with varying densities.

**7. Visualization:** The clustering results were visualized using scatter plots, where clusters were represented by distinct colors and noise points by gray. Visual analysis revealed that KDE-based clustering effectively captured complex patterns and irregularly shaped clusters.

**8. Comparative Analysis with DBSCAN:** A comparative analysis highlighted the advantages of KDE-based clustering over DBSCAN. The flexibility in selecting kernel functions and adaptive bandwidth parameters allowed KDE to handle datasets with varying densities effectively. Unlike DBSCAN, which misclassified overlapping regions as noise, KDE-based clustering provided robust cluster assignments with better precision.

Through this detailed methodology, kernel density-based clustering demonstrates its capability to overcome the limitations of traditional methods, offering a versatile and adaptive approach to clustering in diverse datasets. The numerical results validate the effectiveness of the proposed method, showcasing its potential for future applications in real-world scenarios.

1. REAL WORLD APPLICATION

Explored the application of clustering techniques such as DBSCAN and kernel density-based methods. The dataset used involves geographical or spatial data, potentially tied to crime analysis, as indicated by the clustering focus. Initial exploratory data analysis visualizes the spatial distribution, laying a foundation for DBSCAN and kernel density clustering methods.

A map with blue dots

Description automatically generated

**Fig 8**: Spatial Representation of NYPD Shooting Incident Data 2006 - 2021.

DBSCAN efficiently groups data points based on density thresholds but might struggle with uneven distributions or varying densities. Kernel density-based clustering enhances these results by leveraging customizable kernels (e.g., Gaussian, Epanechnikov) to estimate density more adaptively. This approach improves cluster separation and delineation, particularly in complex or noisy datasets, leading to more nuanced and accurate results.

A map of a city

Description automatically generated

**Fig 9**: Spatial Representation of DBSCAN Clustering on NYPD Shooting Dataset.

A group of maps with different colors

Description automatically generated

**Fig 10**: Spatial Representation of Various Kernel Based Clustering on NYPD Shooting Dataset.

IV. CONCLUSION

This study demonstrates the effectiveness of kernel density-based clustering as a robust alternative to traditional density-based methods like DBSCAN. By leveraging Kernel Density Estimation (KDE) with various kernel functions, this approach overcomes the limitations of fixed parameter dependence and uniform weighting in traditional algorithms. KDE's flexibility allows for adaptive clustering, effectively handling datasets with varying densities and irregular cluster shapes.

The experimental results validate the superiority of kernel density-based clustering, showing improved silhouette scores and more precise cluster formation. The ability to customize kernel functions, such as Gaussian and Epanechnikov, enhances the method's adaptability to diverse datasets. Additionally, this technique exhibits resilience in distinguishing meaningful clusters from noise, a significant challenge for traditional methods.

The findings of this study pave the way for broader applications of kernel-based clustering in domains requiring nuanced data segmentation. The proposed methodology highlights its potential for integration with real-time systems and large-scale datasets, marking a significant advancement in the field of clustering and density estimation. Future research will focus on optimizing computational efficiency, exploring advanced kernel functions, and extending the methodology to high-dimensional spaces.

ACKNOWLEDGEMENT

My heartfelt gratitude to Professor Sung Hyu Cha from Pace University for his guidance and support throughout this research on kernel density-based clustering. His invaluable insights and expertise have greatly enriched our understanding of advanced clustering techniques. I also extend our thanks to the Seidenberg School of Computer Science and Information Systems for providing the resources and encouragement to pursue this work. This study would not have been possible without the inspiration and mentorship of Professor Cha, whose dedication to fostering innovation has been instrumental in our learning journey.

REFERENCES

1. <http://en.wikipedia.org/wiki/Kernel_density_estimation%20>
2. <https://www.sciencedirect.com/science/article/pii/S0957417421002450%20>
3. <https://ieeexplore.ieee.org/document/7803433%20https://link.springer.com/article/10.1007/s10489-022-03754-6%20>
4. <https://www.sciencedirect.com/science/article/pii/S1877050917321750%20http://en.wikipedia.org/wiki/Density-based_clustering%20>
5. Silverman, B. W. (1986). Density Estimation for Statistics and Data Analysis. Chapman & Hall.
6. Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96).
7. Wang, Q., Fan, W., Zhang, J., & Sun, Y. (2021). Adaptive Kernel Density-Based Clustering for High-Dimensional Data. In: International Conference on Computational Intelligence. IEEE.
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.
9. Sheikholeslami, G., Chatterjee, S., & Zhang, A. (1998). WaveCluster: A multi-resolution clustering approach for very large spatial databases. In Proceedings of the 24th International Conference on Very Large Data Bases.