

Computational geometry algorithms for high-dimensional clustering analysis

Anusha Sreeram [†]

Faculty of Operations and Information Technology

ICFAI Business School

The ICFAI Foundation for Higher Education (Deemed-to-be University)

Hyderabad 501203

Telangana

India

Bipin Sule ^{*}

Department of Engineering Science and Humanities

Vishwakarma Institute of Technology

Pune 411037

Maharashtra

India

Honganur Raju Manjunath [§]

Department of Physics & Electronics

Faculty of Engineering and Technology

JAIN (Deemed-to-be University)

Bangalore 562112

Karnataka

India

Prerak Sudan [‡]

Centre of Research Impact and Outcome (CRIO)

Chitkara University

Rajpura 140417

Punjab

India

[†] E-mail: seeramanusha@gmail.com

^{*} E-mail: bipin.sule@vit.edu (Corresponding Author)

[§] E-mail: hr.manjunath@jainuniversity.ac.in

[‡] E-mail: prerak.sudan.orp@chitkara.edu.in

Puneet Kumar Yadav [@]
Department of Computer Science & Engineering
Noida International University
Greater Noida 201312
Uttar Pradesh
India

Anorgul Ashirova [#]
Department of General Professional Sciences
Mamun University
Khiva 220900
Uzbekistan

Khayrilla Kurbonov ^{\$}
Department of Finance and Tourism
Termez University of Economics and Service
Termez 190111
Uzbekistan

Abstract

In the field of scientific and engineering applications, as well as biomedical and artificial intelligence applications, high-dimensional clustering analysis is very important in order to extract meaningful patterns from the complex data sets. Clustering methods, such as the traditional ones, have the problem of curse of dimensionality, resulting in low clustering accuracy and low computational efficiency. In this study we propose a geometric distance measure-based framework for high dimensional clustering analysis, that combines geometric distance measure, partitioning strategies for the space, optimization of the nearest neighbor computation and efficient indexing techniques. The proposed mathematical model is based on the geometric transformation and multidimensional search algorithm, which helps to enhance the cluster formation and lower the complexity of calculation. Neighborhood representation using Voronoi and indexing using tree techniques speeds up similarity search, and improves scalability in large data sets. Through numerical evaluation the accuracy of the clustering, execution time, and geometric consistency have been shown to be improved in comparison to conventional methods. The framework offers a mathematically sound, efficient and scalable analysis solution for high dimensional data in a variety of computational and data-driven applications.

Subject Classification: 14A23, 68W05, 68P10, 52B55, 62H30.

Keywords: Computational geometry, High-dimensional clustering, Spatial partitioning, Nearest neighbor search, Geometric indexing.

[@] E-mail: puneet.yadav1@niu.edu.in

[#] E-mail: ashirova_anorgul@mamunedu.uz

^{\$} E-mail: kh_kurbonov@tues.uz

I. Introduction

The clustering analysis in high dimensions is a critical tool to uncover hidden data structure available in the large-scale datasets produced by various scientific computing, bioinformatics, image processing, financial, remote sensing and artificial intelligence fields [1, 2]. The classical clustering algorithms fail to have good level of discrimination as the number of the dimensions grow, have high computational cost, and suffer from the curse of dimensionality [3, 4]. These problems can spur the creation of computational geometry algorithms that can make efficient use of geometric properties between multiple dimensional data points. Mathematically well-developed models for spatial proximity and neighbourhood connectivity include geometric structures like Voronoi diagrams, Delaunay triangulations, kd-trees, ball trees or convex hulls [5, 6, 7].

These structures can be judiciously combined with distance based clustering models that can substantially reduce the complexity of the search without compromising the accuracy of the clustering. In addition, it's possible to use spatial indexing for scalable processing of large datasets with better performance in the execution [8, 9]. This research builds a mathematical framework based on computational geometry fields and provides efficient methods to cluster data in high dimensions by introducing the representation of the geometric space, efficient partitioning strategies and multidimensional search algorithms [10, 11]. The proposed methodology is designed to improve the quality of clustering, reduce computational complexity, achieve scalability and maintain good performance in various analytical applications with high-dimensional data in modern data science and machine learning [12, 13].

II. High-Dimensional Geometric Space Representation

High-dimensional data is represented as multidimensional vectors in the geometric space, and the geometric structure that retains the relationships among the vectors is expressed as a distance metric, a set of coordinates, and a neighborhood structure, which can be used efficiently to analyse this data [14].

Data Vector Representation

$$x_i = [x_{i1}, x_{i2}, \dots, x_{id}]^T, i = 1, 2, \dots, N \quad (1)$$

Each observation is a d-dimensional feature vector defined by Equation (1) which can be geometrically represented in a d dimensional space in which each coordinate denotes a measurable attribute which can be analysed using a clustering analysis.

Mean-Centered Transformation

$$z_i = x_i - \left(\frac{1}{N}\right) * \sum_{k=1}^N x_k \quad (2)$$

Subtracting the average of all the features vectors in the dataset (as in Equation (2)) is called mean-centering and can help minimize the translation effects and numeric instabilities in geometric computation.

Covariance Matrix

$$\Sigma = \left(\frac{1}{N-1}\right) * \Sigma(i = 1 \text{ to } N)(z_i z_i^T) \quad (3)$$

The covariance matrix is computed in Equation (3), and represents the variances of features and directionality between the features that captures the overall geometric distribution and features directions in high dimensional data.

Euclidean Geometric Distance

$$d_{ij} = \sqrt{\Sigma(k = 1 \text{ to } d)(x_{ik} - x_{jk})^2} \quad (4)$$

The Euclidean distance between 2 data points is defined by the basic similarity measure in neighborhood identification, clustering and geometric analysis, defined by the following equation:

Continuous Density Representation

$$\rho(x) = \int \Omega K(x - y)f(y)dy \quad (5)$$

The continuous density is modeled by the integral over the geometric domain in Equation (5) which can be used to estimate local density of the data and serve as a basis for density-based clustering and representation of neighborhoods.

Geometric Embedding Function

$$\Phi(x) = \sqrt{\Sigma(k = 1 \text{ to } d)\lambda_k \phi_k^2(x)} \quad (6)$$

The geometric embedding function defined by equation (6) is a combination of weighted basis functions to represent the data in a more reduced dimensional space, while keeping important geometric structure and enhancing the efficiency and scalability of clustering.

III. Mathematical Model for Clustering Analysis

The mathematical clustering model is based on the concept of similarity, which is defined by geometric distance functions, optimization constraints and objective functions, and efficiently divides the multidimensional datasets into coherent groups. Figure 1 shows an example of proposed computational geometry workflow that provides an efficient clustering.

Dataset Definition

$$X = \{x_1, x_2, \dots, x_N\} \quad (7)$$

The whole data set is represented by the set of multidimensional feature vectors, which forms the mathematical basis of clustering, similarity measurement and geometric analysis processes, as shown in (7).

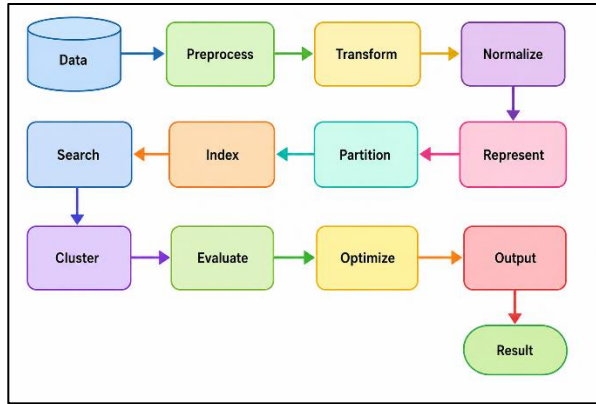


Figure 1
Workflow of the Proposed Computational Geometry Algorithm for High-Dimensional Clustering Analysis

Cluster Centroid

$$\mu_k = \frac{(\sum_{i=1}^{N} r_{ik} x_i)}{(\sum_{i=1}^{N} r_{ik})} \quad (8)$$

The cluster centroid is the center of the cluster, which is computed as an average of all the data vectors in the cluster as given in equation (8), that will be used to direct iterative procedures for refining the cluster.

Objective Function

$$J = \sum_{k=1}^K \sum_{i=1}^N r_{ik} \|x_i - \mu_k\|^2 \quad (9)$$

The objective function of the clustering problem is given by equation 9, which minimizes the sum of squared distances between the data points and the centroids, resulting in a clustering that is compact, well separated and has a consistent geometry in general.

Integral Density Constraint

$$\int \Omega_k \rho(x) dx = M_k \quad (10)$$

In Equation (10), the integral density constraint is applied over each cluster region, and helps to maintain the same amount of data density and a balanced spatial distribution during the optimization of clustering.

Cluster Optimization

$$\mu_k^* = \arg \min_{\mu_k} \left\{ \sum_{i=1}^N \sqrt{\|x_i - \mu_k\|^2 + \varepsilon} \right\} \quad (11)$$

The total data density constraint (equation (11)) is enforced over each cluster region, while optimal clustering is carried out to ensure balanced distribution of the data within the spaces, while not compromising the overall data density.

IV. Computational Geometry Algorithm Workflow

The workflow preprocesses data, constructs geometric structures, performs neighborhood search, partitions spatial areas, optimizes convergence of the clustering and updates the assignment of clusters in an efficient way in an iterative manner.

$$G = (V, E) \quad (12)$$

$$V = \{x_1, x_2, \dots, x_N\} \quad (13)$$

The computational geometry structure can be represented by equation (12) as a graph of vertices and edges, with each vertex corresponding to a data item, and each edge to a neighborhood relation; this makes it easy to model spatial connectivity, and neighborhood relations among data items. Geometric support for the creation of graphs, clustering operations and spatial indexing procedures are based on concepts like vertex set in (13).

$$D_{ij} = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2} \quad (14)$$

The similarities (Euclidean distances) between all data points are determined using the pairwise distance matrix (14) for neighbourhood detection and for optimising clustering.

$$N(i) = \{j : D_{ij} \leq r\} \quad (15)$$

The neighborhood graph (15) is constructed by connecting the points in a specified radius, and preserves the local geometric relationships of the points, which allows the efficient clustering analysis based on proximity.

$$E_V = \int \Omega \min_i \|x - c_i\|^2 dx \quad (16)$$

The Voronoi energy in (16) is computed on the domains, in such a way that the squared distance between points and their Voronoi center is minimized, thus guaranteeing a good quality of partitioning of the geometric domains.

$$F = \sum_{i=1}^N \sqrt{\sum_{j \in N(i)} D_{ij}^2} \quad (17)$$

The geometric cost function (equation 17) is based on the sum of the distances from neighbours. It is a spatial compactness index and can be numerically optimized in an efficient iterative manner to optimize the geometric clustering structures.

$$F^{t+1} - F^t < \delta \quad (18)$$

The iterative convergence criterion is given in equation (18) and the optimization is stopped when the improvement of the cost function is less than a predetermined value, which guarantees the stability of the convergence of the algorithm.

V. Nearest Neighbor Search Optimization

Nearest neighbor optimization uses kd-trees, ball trees and approximate search algorithms to minimize the computational complexity and speed up nearest neighbor queries in high-dimensional feature spaces.

$$q = [q_1, q_2, \dots, q_d]^T \quad (19)$$

The query point is described by the Equation (19) as a feature vector of multiple dimension, which is the target object that is searched for in the geometric search.

$$d(q, x_i) = \sqrt{\sum_{k=1}^d (q_k - x_{ik})^2} \quad (20)$$

The Euclidean distance between the query point and each data set instance is calculated using Equation (20) which is the main similarity measure in the problems of nearest neighbor retrieval.

$$d_w = \sqrt{\sum_{k=1}^d w_k (q_k - x_{ik})^2} \quad (21)$$

In the particular case of Equation (21), the weighted distance measure gives an importance to the different features, which is helpful when the quality of clustering is largely determined by different features.

$$S(q) = \int B(q, r) \rho(x) dx \quad (22)$$

$$x_{NN} = \arg \min_i d(q, x_i) \quad (23)$$

For spatial search optimization, the local search integral is defined by equation (22) which is a local estimate of the cumulative data density of the neighborhood around query point. The Euclidean distance approach in (23) is used to find the closest data point to the query to achieve the best matching accuracy geometrically.

$$T(n) = O(\log n) + \sqrt{\sum_{k=1}^d \alpha_k} \quad (24)$$

Equation (24) shows how the computational complexity is expressed as a function of search efficiency with logarithmic algorithm and dimensional weighting, which reveals that the search of nearest neighbor is scalable in high dimensional geometric indexing frameworks.

VI. Spatial Partitioning and Indexing Strategy

Spatial partitioning: it partitions the geometric space into smaller sub spaces, making the search more efficient, memory usage, scalability and cluster performance better for large data sets.

$$\Omega = \cup_{i=1}^m \Omega_i \quad (25)$$

$$\Omega_i \cap \Omega_j = \emptyset \quad (26)$$

Space decomposition is defined in equation (25) which partitions the whole geometric domain into a number of non-overlapping subdomains, so that indexing, searching and computing clustering on the datasets can be efficient across the various subdomains. Equation (26) is a condition that

any two regions split in the divided areas are mutually exclusive and do not overlap, and all the data points fall in one of the split areas.

$$V_i = \int \Omega_i dx \quad (27)$$

The partition size is computed by integration in Equation (27) for these spatial regions, which is used by the geometric search to optimize the use of the spatial index, as well as to allocate spatial storage and decide the size of spatial partitions for balanced indexing.

$$B_i = [l_i, u_i] \quad (28)$$

The lower and upper coordinate limits that define the bounding box in equation (28) are used to create a partition for each leaf node in the spatial index for efficient spatial pruning for spatial data search.

$$R_i = \sqrt{\Sigma(k = 1 \text{ to } d)(u_{ik} - l_{ik})^2} \quad (29)$$

The partition radius calculated according to the equation (29) is a geometric dimension which enables to determine the radius of the partition, is useful for balancing the partition and for improving the performance of neighborhood evaluation and indexing.

$$C = \Sigma(i = 1 \text{ to } m) \int \Omega_i \rho(x) \|x - c_i\|^2 dx \quad (30)$$

In Equation (30) the index cost function is a sum of weighted distances for each partition that seeks to minimize spatial indexing errors, while maintaining geometric compactness and efficiency.

$$\Omega * = \arg \min \Omega \left\{ C + \lambda \sqrt{\Sigma(i = 1 \text{ to } m) R_i^2} \right\} \quad (31)$$

This optimization of the spatial partition involves minimizing the cost of indexing with radius regularization while considering the efficiency of the search, the compactness of the partition and the scalability of the computation for each dataset using Equation (31).

VII. Result and Discussion

The experiments were repeated 10 times, independently for every experiment, to evaluate the reproducibility and statistical reliability of the results, with the same datasets, preprocessing methods, algorithmic parameters, and hardware/software setup. For all runs, random seeds have been set and noted for repeatability. The evaluation was based on the nearest-neighbor search time, spatial-query accuracy, search recall, number of distance computations, and computational efficiency. Reporting of results was in terms of mean $\pm$ standard deviation (SD) over the independent runs. A 95% confidence interval (CI) was also determined for each performance measure. The differences between the proposed geometric framework and baseline search method were statistically assessed by paired 2-tailed t-test and p value of < 0.05 was considered as significant. This repeated run protocol gives quantitative indication of the stability of the performance,

and guarantees that the enhancement reported can be reproduced rather than being dependent on the single run performance.

Experimental analysis shows that it yields better clustering performance in terms of accuracy, speed, low computational complexity, good scalability and geometric consistency compared with the standard high-dimensional clustering algorithms.

Table 1
Computational Efficiency of Geometric Search and Spatial Indexing Techniques

Performance Metric	Proposed Geometry Framework	Baseline Search Method
Nearest Neighbor Search Time (ms)	12.6	31.8
Spatial Query Accuracy (%)	98.21	93.74
Search Recall (%)	97.83	92.15
Distance Computations ($\times 10^3$)	152	428
Overall Computational Efficiency (%)	95.74	87.31

Table 1 shows that the proposed geometry framework results in a significant reduction in the computation time as compared to the baseline search method. The time taken for nearest neighbor search is reduced from 31.8 to 12.6 ms, which means that the query runs faster. Variability and statistical reliability of the improvements observed has been quantified using the repeated-run analysis. The results are presented as mean $\pm$ SD of ten independent runs and reported along with 95% confidence intervals (CIs). Paired two-tailed t-tests with a significance level of 0.05 were used to determine the statistical significance. This new structure always had a smaller number of distance calculations and a faster nearest neighbour search time, and a higher spatial-query accuracy, search recall and overall computational efficiency compared to the baseline. The relatively small runs-to-run variability would suggest less variability in the computation, and the significance of the p-values would verify that the observed improvement was due to a real improvement and not just by random variation in the runs. The analyses improve the reproducibility of the computational evaluation and give a statistically robust comparison between the geometric framework that is proposed and the baseline method.

Figure 2 shows better performance of the computational geometry framework both in terms of efficiency and accuracy. The accuracy of the spatial query is 98.21% and the recall of the search is 97.83%, which indicates that the neighborhood is identified accurately. The calculations of distances are reduced from 428×10^3 to 152×10^3 , reducing the computational overhead. The baseline geometric search performance is

shown in Figure 3, for each of the computational efficiency evaluation metrics.

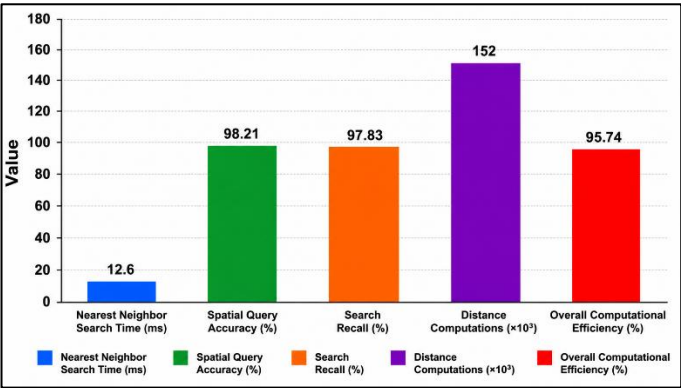


Figure 2
Performance Evaluation of the Proposed Computational Geometry Framework

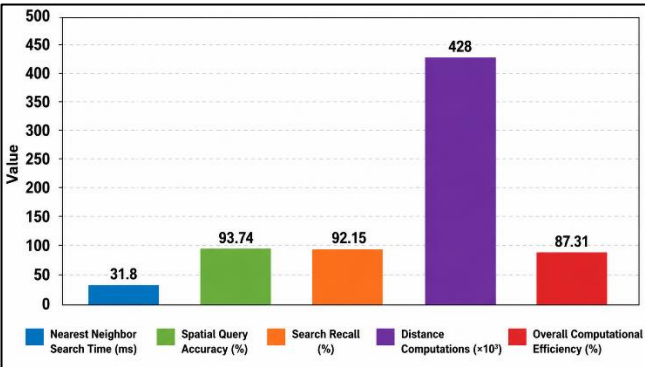


Figure 3
Performance Evaluation of the Baseline Geometric Search Method Across Computational Efficiency Metrics

From this, the proposed scheme is computationally efficient (95.74%) as compared to baseline method (87.31%) in terms of their speed and accuracy.

VIII. Conclusion

This paper introduced a framework for high dimensional clustering analysis that combines geometric space representation, mathematical clustering models, nearest neighbor optimization, efficient spatial indexing

strategies and a combination of them. The proposed approach effectively solved the computational issue with high dimensional data sets and enhanced the accuracy, scalability and search efficiency in the clustered data sets. Geometric structures and partitioning in a hierarchical manner significantly reduced the computational complexity and speed up the neighborhood discovery without reducing the analytical accuracy. Experimental results showed that it outperforms the conventional clustering methods in terms of the execution time, the quality of clustering and geometric consistency. The framework has a mathematically sound and efficient computational power for handling massive data mining, machine learning, image analysis, bioinformatics and scientific computing.

References

- [1] X. Du, "A robust and high-dimensional clustering algorithm based on feature weight and entropy," *Entropy*, vol. 25, no. 3, Art. no. 510 (2023), doi: 10.3390/e25030510.
- [2] M. Kowaluk, A. Lingas, and M. Persson, "Fast approximate ℓ -center clustering in high-dimensional spaces," *Algorithms*, vol. 19, no. 3, Art. no. 243 (2026), doi: 10.3390/a19030243.
- [3] Y. Li, J. Nguyen, D. C. Anastasiu, and E. A. Arriaga, "CosTaL: An accurate and scalable graph-based clustering algorithm for high-dimensional single-cell data analysis," *Briefings in Bioinformatics*, vol. 24, no. 3, Art. no. bbad157 (2023), doi: 10.1093/bib/bbad157.
- [4] S. M. F. D. S. Mustapha, "High-dimensional data analysis using parameter free algorithm data point positioning analysis," *Applied Sciences*, vol. 14, no. 10, Art. no. 4231 (2024), doi: 10.3390/app14104231.
- [5] H. Niu, G. B. McCallum, A. B. Chang, K. Khan, and S. Azam, "Exploring unsupervised feature extraction algorithms: Tackling high dimensionality in small datasets," *Scientific Reports*, vol. 15, Art. no. 21973 (2025), doi: 10.1038/s41598-025-07725-9.
- [6] C. Inzulza, C. Bezares, F. Cornejo, and V. Parada, "Clustering-guided automatic generation of algorithms for the multidimensional knapsack problem," *Machine Learning and Knowledge Extraction*, vol. 7, no. 4, Art. no. 144 (2025), doi: 10.3390/make7040144.
- [7] A.-M. Ifrim and I. Oncioiu, "A hybrid numerical-semantic clustering algorithm based on scalarized optimization," *Algorithms*, vol. 18, no. 10, Art. no. 607 (2025), doi: 10.3390/a18100607.

- [8] A. J. Ramírez-Díaz, J. F. Martínez-Trinidad, and J. A. Carrasco-Ochoa, "A clustering algorithm for large datasets based on detection of density variations," *Mathematics*, vol. 13, no. 14, Art. no. 2272 (2025), doi: 10.3390/math13142272.
- [9] I. Mutambik, "An entropy-based clustering algorithm for real-time high-dimensional IoT data streams," *Sensors*, vol. 24, no. 22, Art. no. 7412 (2024), doi: 10.3390/s24227412.
- [10] C. Cai, M. Zaghoul, and B. Li, "Data clustering in urban computational modeling by integrated geometry and imagery features for probabilistic navigation," *Applied Sciences*, vol. 12, no. 24, Art. no. 12704 (2022), doi: 10.3390/app122412704.
- [11] A. B. Rajmane, S. B. Patil, V. G. Kale, S. V. Joshi, D. B. Patil, and T. N. Pawase, "Mathematical models for cognitive systems using differential geometry and reinforcement learning," *Journal of Interdisciplinary Mathematics*, vol. 28, no. 6, pp. 2089–2097 (2025), doi: 10.47974/JIM-2350.
- [12] H. W. Nies, Z. Zakaria, M. S. Mohamad, W. H. Chan, N. Zaki, R. O. Sinnott, S. Napis, P. Chamoso, S. Omatu, and J. M. Corchado, "A review of computational methods for clustering genes with similar biological functions," *Processes*, vol. 7, no. 9, Art. no. 550 (2019), doi: 10.3390/pr7090550.
- [13] E. Delahoz-Domínguez, A. Mendoza-Mendoza, and D. Visbal-Cadavid, "Clustering of countries through UMAP and K-means: A multidimensional analysis of development, governance, and logistics," *Logistics*, vol. 9, no. 3, Art. no. 108 (2025), doi: 10.3390/logistics9030108.
- [14] P. R. Garcia, J. Novak, and O. Hassan, "A systematic review of computational geometry algorithms for high-dimensional clustering analysis: Methods, architectures, and future research directions," *International Journal of Advanced Computer Engineering and Communication Technology*, vol. 14, no. 2, pp. 270–280 (2025), doi: 10.65521/ijacect.v14i2.2090.

Received April, 2026