**COMP 8157 Advanced Database Topics**

**Milestone Report**

**Group Name:** Vector Core

**Project Title:** Optimizing Vector Database Retrieval through Adaptive Hierarchical Clustering

**Contribution:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Student ID** | 110167386 | 110163138 | 110185214 | 110175870 | 110177924 | 110180669 |
| **Student Name** | Kathan Piyushkumar Bhavsar | Dev Pinkainbhai Patel | Mohammad Arsh Vahora | Kalp Jigneshbhai Kalani | Kush Janak Patel | Hetanshu Hemant Patel |
| **Contribution** | 16.67 | 16.66 | 16.66 | 16.66 | 16.66 | 16.66 |
| **Signature** |  |  |  |  |  |  |

**CONFIDENTIALITY AGREEMENT & STATEMENT OF HONESTY**

We, verify that the submitted work is my own, original work, that all sources are cited accurately, and that I have not submitted any portion of this work for any other university course. We did not use Generative AI tools (e.g., ChatGPT, Bard) to produce this assignment or report. We confirm knowing that a mark of 0 may be assigned for copied work.

\_\_\_\_\_\_\_\_\_\_\_ Kathan Piyushkumar Bhavsar\_ 110167386



Student Signature Student Name (please print) Student I.D. Number

\_\_\_\_\_ Dev Pinakinbhai Patel\_\_\_\_\_\_\_ 110163138



Student Signature Student Name (please print) Student I.D. Number

\_\_\_\_\_\_ Mohammad Arsh Vahora\_\_\_\_ 110185214



Student Signature Student Name (please print) Student I.D. Number

\_\_\_\_\_\_ Kalp Jigneshbhai Kalani\_\_\_\_\_\_ 110175870



Student Signature Student Name (please print) Student I.D. Number

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Kush Janak Patel\_\_\_\_\_\_\_\_\_ 110177924



Student Signature Student Name (please print) Student I.D. Number

\_\_\_\_\_\_\_ Hetanshu Hemant Patel\_\_\_\_ 110180669



Student Signature Student Name (please print) Student I.D. Number

Optimizing Vector Database Retrieval through Adaptive Hierarchical Clustering

Kathan Piyushkumar Bhavsar   
*University of Windsor*  
bhavsa85@uwindsor.ca

Kalp Jigneshbhai Kalani  
*University of Windsor*kalanik@uwindsor.ca Dev Pinakinbhai Patel  
*University of Windsor*patel41b@uwindsor.ca

Kush Janak Patel  
*University of Windsor*patel9ib@uwindsor.ca Mohammad Arsh Vahora  
*University of Windsor*vahora4@uwindsor.ca

Hetanshu Hemant Patel  
*University of Windsor*patel8qb@uwindsor.ca

*Abstract*— We propose an improved approach for vector similarity search optimization based on an enhanced hierarchical clustering system. The current solutions have three main flaws, such as efficiency in database restructuring, feature preservation, and multi-cluster membership handling. Our proposed system introduces adaptive parameters for controlled restructuring, weighted centroid calculation, and a skip algorithm for multi-cluster elements in order to reduce the search complexity with quality maintenance of results and address maintenance challenges in the vector databases. It integrates three key novel solutions: adaptive restructuring mechanism, context-preserving centroid computation, and skip algorithm for multi-cluster management; all together, these provide a more efficient and maintainable vector database system.

Keywords — vector database, hierarchical clustering, similarity search

# Introduction

Modern applications ranging from recommendation systems to semantic search engines have been integrating vector databases. These databases store and process high-dimensional vectors representing complex data, such as images, text embeddings, and user behaviour patterns. However, current implementations are afflicted with several critical challenges that limit their effectiveness, especially considering the ever-increasing scale and complexity of applications [2].

The challenges are mainly centered on search efficiency, where the need to compare the query vectors with every first-level centroid creates a huge computational bottleneck. Since traditional methods require exhaustive comparisons even for preliminary clustering steps, these have linear time complexity concerning database size [4]. This problem scales up with the scaling up of databases, hence resulting in poor performance in large-scale applications. This leads to the fact that in real-world datasets, the query time could be above the acceptable threshold of interactive applications, particularly those systems relying on real-time answers [3].

Another major challenge is contributed by the maintenance overhead, as most of the existing hierarchical methods have to be remade in their entirety in case of new insertions. The addition of new vectors in current systems is many times achieved at the cost of going through several layers of hierarchy again for optimal clustering, which has significant computational cost and takes down the system from availability [5]. The problem is worse in dynamic systems, where updates should be very frequent, since every update operation can start expensive reorganization processes in the whole database structure [5].

At higher levels of granularity, feature dilution occurs due to simple averaging while calculating centroids representative of meaningful features and context of the original vectors. Moving deeper in the hierarchy, each higher level of abstraction loses more of the distinctive characteristics that make the original vectors meaningful; hence, potential loss in precision due to cluster overlap may deteriorate the quality of the search results [1]. This effect is amplified for applications in which small-scale vector differences carry important semantics, such as semantic search or recommendation systems, where subtle relationships between items must be preserved [3].

Current systems also have difficulty dealing with multi-cluster membership, and often one has to make arbitrary assignments for vectors that rightfully belong to multiple clusters. Many real-world entities naturally share characteristics with multiple groups, but traditional clustering approaches typically force each vector into a single cluster. This may seriously affect retrieval accuracy and is not representative of the natural relationships in data [4]. In such cases, for instance, the content recommendation system would classify a movie either under "action" or "comedy", while such a movie might belong to both categories. In this case, the recommendations are either incomplete or totally wrong [3].

In this regard, exponential growth in size and utilization of vector databases demands further scalable search methods. Applications keep generating more and processing increased volumes of vector data, thereby giving rise to emergent needs concerning the efficient mechanism for searching. Most of the current solutions cannot maintain acceptable performance with growing data [2]. There lies great potential for optimization in the computational overhead during database updates while preserving cluster quality. This creates the bottleneck for executing restructuring operations at high costs for real-time applications in current systems, particularly dynamic ones that do need updates quite frequently [5].

Preserving vector semantics across hierarchical levels is central to achieving meaningful and precise search results. The loss of semantic information in current systems leads to decreased search precision and reduced overall system effectiveness [1]. The accommodation of natural multi-cluster memberships is of growing importance for many real-world applications. Many contemporary applications require vectors to belong to multiple clusters, making this capability essential in truly representing the data and performing accurate retrieval [4].

Our proposed solution meets these challenges by means of three key innovations.

Adaptive Restructuring System introduces user-controlled parameters in order to give the power of fine-tuning to database maintenance. It basically allows the setting of a tree height to be configured as per customized hierarchical depth; hence, a user can make a trade-off between the speed of search versus memory use. The system continuously monitors cluster sizes through a rechecking factor that triggers reorganization only when necessary. It also applies selective tree reorganization using new cluster factors so that the restructuring operations will affect only the affected clusters and not the whole database. This has massively reduced the maintenance overhead while still enabling users to change parameters w.r.t. their performance requirements and needs for stability [5].

Context-Preserving Centroid Calculation introduces an efficient way of maintaining semantic relevance along the hierarchy. This method applies the inverse multiplying weights in the calculation and allows dynamic weight adaptation concerning the similarity distances between vectors. It gives more importance to the contribution at the border for giving a better edge to each cluster and retaining semantic boundaries [8]. Iteratively, the position of centroids is optimized in order for them to represent their cluster best. This sophisticated approach maintains semantic relevance at higher levels of hierarchy, hence giving more accurate search results and better cluster representation across the entire database structure [4].

The Skip Algorithm for Multi-Cluster Management proposes a two-phase clustering scheme to manage complex vector relationships. The system identifies and temporarily skips the vectors showing significant similarity in multiple clusters without forcing them to immediate assignment during the first phase of clustering. The secondary clustering process of these skipped vectors analyzes the detailed relation with the previous clusters. Finally, the system will perform multi-cluster assignment on vectors that may have remained ambiguous, allowing the vectors to point to multiple clusters to which they relate [8]. This organized way of treating vectors that naturally belong to several clusters significantly enhances the overall accuracy of the system.

Together, these components present a complete package for tackling core problems that come with most state-of-the-art vector database solutions. Equipped with an adaptive restructuring system, context-preserving calculation method, and with multi-cluster sophisticated handling, the proposed system performs with increased effectiveness and quality while being able to provide efficiency during real-life applicability.

This is a proposed solution that takes advantage of combining advanced toolchains and technologies in order to achieve better performance for functionalities within vector databases. The Python-based frameworks include an adaptive restructuring system, a context-preserving centroid computation, and a skip algorithm for managing multiple clusters-all built on NumPy for vector operations and SciPy for clustering algorithms. Machine learning libraries such as PyTorch are further used to dynamically adjust weights and optimize centroids so that semantic relevance can be maintained in high-dimensional data.

# Literature Study

The recent upsurge of high-dimensional data in modern applications, such as recommendation systems and semantic search engines, has pushed the need for efficient and scalable vector database retrieval methods. Traditional hierarchical clustering approaches are likely to suffer from search efficiency, maintenance overhead, semantic preservation, and multi-cluster membership. This literature review critically examines existing methods, their advantages, limitations, and delineates how the proposed research aims to address these shortcomings.

## Hierarchical Clustering in Vector Databases

Nishant et al. [1] implemented an effective information retrieval system using hierarchical clustering to enhance query effectiveness. Their system successfully minimized the search space by organizing the similar documents in a hierarchical structure. However, the necessity to recalculate cluster structures at every update incurred large maintenance overhead and therefore was not very efficient for dynamic systems with continuous data insertion.

Chen et al. [2] presented SingleStore-V, a combined vector database system that combines conventional database indexing with vector similarity search. The hybrid indexing technique maximized nearest-neighbor searches but failed to completely eliminate scalability problems inherent in hierarchical clustering, particularly in high-dimensional data environments.

Kale [3] investigated the disruptive effect of vector databases on content personalization. The research highlighted difficulties in identifying nuanced semantic relations between vectors, pointing out that conventional clustering algorithms tend to dilute features at higher levels of abstraction, hence degrading the accuracy of search results.

Naumov et al. [4] suggested an objective-based hierarchical clustering technique specific to deep embedding vectors. Although their method enhanced the efficiency of clustering deep-learning-generated embeddings, it imposed rigid single-cluster membership. This restriction precluded the ability to represent vectors that inherently have multiple memberships, impacting retrieval precision.

## Advances in Adaptive Hierarchical Clustering

Emamjomeh-Zadeh and Kempe [9] introduced an adaptive hierarchical clustering algorithm with ordinal queries to compute element proximities. Their method constructs hierarchical clusterings with a logarithmic number of adaptive queries, and it is also resilient even in noisy environments. However, application of ordinal data may limit applicability in cases requiring precise distance measures.

Eriksson et al. [10] proposed an active clustering framework that select pairwise similarities adaptively to construct hierarchical clusterings efficiently. By retaining informative similarities only, their method reduces computational requirements and hence can be scaled. However, the assumption that intracluster similarities are always higher than intercluster similarities does not work for all datasets.

Deolalikar and Laffitte [11] designed an adaptive hierarchical clustering algorithm that iteratively improves clusters based on quality measures. Subclusters not meeting pre-established thresholds are partitioned, thereby increasing the precision of the clustering. While efficient, the method may be at the cost of increased computational complexity with multiple clustering iterations.

## Vector Database Retrieval and Indexing

Jin et al. [12] introduced Curator, an index scheme for multi-tenant vector databases. Curator employs tenant-specific clustering trees concisely represented in shared structure, optimizing memory efficiency for query performance at the cost. The system complexity may be tough in dynamic environments with high rates of tenant update.

Pham and Do [13] presented an indexing method that integrated hierarchical clustering and product quantization for large-scale object retrieval. With subspace decomposition and the use of hierarchical clustering in each subspace, the method preserves efficient and effective search performance. The primary limitation is the potential loss of global feature relationships due to subspace partitioning.

Vijayan and Aziz [14] proposed an Adaptive Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm tailored to stream data. The process modifies the clustering in a dynamic fashion to accommodate moving data points, enhancing execution time at the cost of no compromise in accuracy. However, the technique primarily caters to spatial data, and the applicability of the process on other types of data might be restricted.

## Limitations in Existing Approaches

Although hierarchical clustering and vector database retrieval have witnessed colossal advancements, several challenges remain. One of the principal challenges is computational overhead because approaches like adaptive clustering [10] and iterative refinement [11] are computationally costly, hence ineffective when dealing with large-scale data. Another challenge is maintenance complexity, particularly in systems like Curator [12] that fail to support dynamic environments nicely where data along with tenant structures evolve rather frequently. One is semantic preservation because subspace partitioning-based methods [13] can lead to the loss of global semantic relationships at the expense of retrieval accuracy. Another is that most clustering algorithms enforce exclusive cluster membership, which fails to adequately account for data points that naturally belong to multiple clusters. It is overcoming these limitations that are critical in taking vector database retrieval systems to higher efficiency, scalability, and accuracy.

# Proposed Model

##### References

1. P. S. Nishant, S. Mehrotra, P. R. Sree and P. Srikanth, "Hierarchical clustering based Intelligent Information Retrieval approach," *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2020, pp. 862-866. <https://ieeexplore.ieee.org/abstract/document/9214231?casa_token=JTIN4_SZ4pYAAAAA:Vb2i77RGDiMJm4Vusl1_5HWvx2nCXXtiuQzFkA5feLqycuqv_-gu069sZd5C3gkVKaQXIVYDdf0>
2. C. Chen, C. Jin, Y. Zhang, S. Podolsky, C. Wu, S. Wang, E. Hanson, Z. Sun, R. Walzer, and J. Wang. 2024. “SingleStore-V: An Integrated Vector Database System in SingleStore.” Proc. VLDB Endow. 17, 12 August 2024, 3772–3785. <https://doi.org/10.14778/3685800.3685805>
3. S. Kale, “From Text to Recommendations: How Vector Databases are Revolutionizing Personalized Content Delivery,” 2024. International Journal for Research in Applied Science & Engineering Technology (IJRASET). vol. 12, April 2024, ISSN: 2321-9653. <https://www.researchgate.net/profile/Satyanand-Kale/publication/380208131_From_Text_to_Recommendations_How_Vector_Databases_are_Revolutionizing_Personalized_Content_Delivery/links/6697340d8dca9f441b82f3f2/From-Text-to-Recommendations-How-Vector-Databases-are-Revolutionizing-Personalized-Content-Delivery.pdf>
4. S. Naumov, G. Yaroslavtsev, and D. Avdiukhin, “Objective-Based Hierarchical Clustering of Deep Embedding Vectors”, *AAAI*, vol. 35, no. 10, pp. 9055-9063, May 2021. <https://ojs.aaai.org/index.php/AAAI/article/view/17094/16901>
5. A. Ponomarenko, Y. Malkov, A. Logvinov, and V. Krylov. "Approximate nearest neighbor search small world approach." In International Conference on Information and Communication Technologies & Applications, vol. 17. 2011. <https://www.iiis.org/CDs2011/CD2011IDI/ICTA_2011/PapersPdf/CT175ON.pdf>
6. Y. A. Malkov and D. A. Yashunin, "Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 4, pp. 824-836, 1 April 2020 <https://ieeexplore.ieee.org/abstract/document/8594636>
7. N. Monath, K. A. Dubey, G. Guruganesh, M. Zaheer, A. Ahmed, A. McCallum, G. Mergen, M. Najork, M. Terzihan, B. Tjanaka, Y. Wang, Y. Wu. 2021. Scalable Hierarchical Agglomerative Clustering. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’21), August 14–18, 2021 <https://dl.acm.org/doi/pdf/10.1145/3447548.3467404>
8. James C. Bezdek, Robert Ehrlich, William Full. “FCM: The fuzzy c-means clustering algorithm”. Computers & Geosciences, Volume 10, Issues 2–3, 1984, Pages 191-203, ISSN 0098-3004, <https://doi.org/10.1016/0098-3004(84)90020-7>.
9. E. Emamjomeh-Zadeh, and D. Kempe. "Adaptive Hierarchical Clustering Using Ordinal Queries." 2018. Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, 415-429.
10. B. Eriksson, G. Dasarathy, A. Singh, and R. Nowak. "Active Clustering: Robust and Efficient Hierarchical Clustering using Adaptively Selected Similarities." 2011. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 260-268.
11. V. Deolalikar, and H. Laffitte. (2015). "Adaptive Hierarchical Clustering Algorithm." U.S. Patent 9,020,271.
12. Y. Jin, Y. Wu, W. Hu, B. M. Maggs, X. Zhang, and D.Zhuo. "Curator: Efficient Indexing for Multi-Tenant Vector Databases. 13, January 2024. <https://arxiv.org/abs/2401.07119>
13. Pham, TA., Do, NT. Embedding hierarchical clustering in product quantization for feature indexing. Multimed Tools Appl 78, 9991–10012 (2019). <https://doi.org/10.1007/s11042-018-6626-9>
14. D. Vijayan, and I. Aziz. Adaptive Hierarchical Density-Based Spatial Clustering Algorithm for Streaming Applications. Telecom. 2023; 4(1):1-14. <https://doi.org/10.3390/telecom4010001>