**Vector Indexing: A Journey into Information Retrieval at Scale [Part 1]**

A network of dots and lines

AI-generated content may be incorrect.

**By Nithin Sai**

**Introduction**

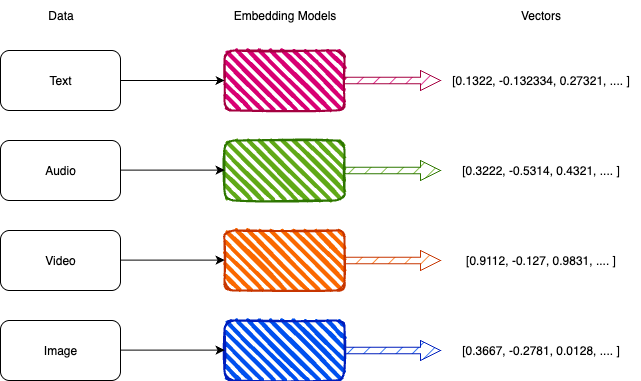
The growing popularity of high-dimensional vector representations is changing the way we approach information retrieval and is reshaping the basic principles of search. In natural language processing, computer vision, and audio analysis, objects are mapped into numerical spaces where semantic similarity is reflected in geometric proximity. Retrieval, therefore, reduces to finding nearest neighbors in these spaces. While conceptually simple, the computational demands at scale are significant.

This article examines the motivation, theoretical background, and necessity of approximate methods in vector indexing. A subsequent article will address practical industry implementations such as IVF, HNSW, and DiskANN.

**Representation as Vectors**

A text document, an image, or an audio signal can be transformed into a vector. For text, embeddings may be obtained from word frequency analysis or neural models. Images may be represented through convolutional feature extraction. Audio signals can be transformed into spectral components. The result is a point in a high-dimensional space, often containing hundreds or thousands of components.

Similarity is then computed geometrically. The Euclidean distance is a standard metric, though cosine similarity and other distance functions may be more appropriate in specific cases. Thus, the abstract problem of semantic retrieval becomes a problem of geometry.



**The Nearest Neighbour Problem**

The fundamental task is the nearest neighbour search. Formally, given a dataset of *n* vectors in a *d*-dimensional space, and a query vector *q*, the objective is to find the stored vector closest to *q* according to a chosen distance function.

The brute-force method compares *q* against all *n* vectors, with complexity O(n·d). Halder et al. (*Journal of Big Data*, 2024, p. 4) confirm this cost, noting that it becomes impractical for large *n*. For example, with a dataset of 10⁹ vectors, each of dimension 512, exhaustive search would require 5×10¹¹ operations per query, beyond reasonable limits for interactive systems.

**The Curse of Dimensionality**

High-dimensional spaces present unique challenges. Distances between vectors tend to concentrate, reducing the contrast between nearest and farthest neighbors.  With higher dimensions, it becomes harder to perform a brute-force naive approach, both because of the cost and also because they become less discriminative. There is a term coined for this exact situation

- *The curse of dimensionality*.

This curse demands different approaches that exploit the fundamental structure of the data rather than perform a uniform 1-to-1 comparison.

**Approximation as Principle**

Approximate Nearest Neighbor Search (ANNS) methods help address the challenges of exhaustive comparisons by focusing on efficiency. They do this by narrowing down the candidate set, but this can sometimes mean missing the exact nearest neighbor.

A systematic review by Halder et al. (2024, pp. 6–7) shows that these Approximation approaches reduce the search time in many orders of magnitude while keeping an accuracy of over 90-95%.

When using approximation, there are certain parameters that control the trade-off between accuracy and performance. These parameters need to be tuned to every dataset; they are specific and very native. There does not exist a single configuration that suits every dataset. This is the case with every **ANN** method.

**Peer-Reviewed Studies**

* **Halder et al. (2024):** Comprehensive survey of k-nearest neighbor methods and modifications. Confirms linear cost of brute-force methods, documents recall-performance trade-offs, and consolidates developments from 1967 to 2022. DOI: [[10.1186/s40537-024-00973-y](https://rdcu.be/eBAca)].

**Conclusion**

Vector indexing arises from the need to perform similarity search in high-dimensional spaces at scale. Exact methods are computationally intractable. Approximate approaches offer a practical resolution, supported by peer-reviewed evidence of their effectiveness. This article has outlined the conceptual and theoretical motivation. The second part, will examine how these principles are realized in practice, focusing on three representative systems: IVF, HNSW, and DiskANN.

**Author**

Nithin Sai is a Software Developer on the DB2 Kernel team at IBM Dublin Lab with a Master's in Computing and Artificial Intelligence from Dublin City University.