Towards Efficient Vector Similarity Search in Analytical Databases

Simon van Noort [1], Leonardo Kuffo [1], and Peter Boncz [1]

[1] Database Architectures, Centrum Wiskunde & Informatica, The Netherlands
{sljvn,lxkr,boncz}@cwi.nl

Vector similarity search (VSS) is commonly used in recommender systems, multimedia retrieval, and chatbots. VSS allows a system to answer questions such as: *What are the five images most similar to the given image?* Approximate vector search indexes make it feasible to process such queries on large datasets within low latency bounds. The state-of-the-art vector index is the graph-based HNSW index. Although HNSW indexes provide low search latency, they have a large memory footprint and are slow to construct and update. Furthermore, the graph-like structure of HNSW makes it difficult to deeply integrate into analytical DBMSs. Fortunately, a recent VSS framework called PDXearch [1] has shown that lightweight partitioning-based IVF indexes can match HNSW index search latency while reducing the index construction times and memory requirements by an order of magnitude.

To bridge the gap between vector similarity search and analytical databases, we are working on a DuckDB extension that integrates PDXearch in DuckDB to efficiently answer VSS queries. This will deliver the community a fast, lightweight, and portable VSS solution, and demonstrates PDXearch's potential at the DBMS level.

An area where deep integration of vector indexing and a DBMS is important is *filtered* VSS [2]. Filtered VSS combines vector similarity search with traditional SQL predicates, and can thus answer the more specific question: *What are the five images most similar to the given image, that were uploaded in 2025?* Analytical DBMSs, such as DuckDB, provide fast predicate evaluation. This opens opportunities to provide fast predicate-agnostic filtered VSS as part of our extension. Recent literature shows multiple strategies to combine predicate evaluation and IVF index search. As a second contribution, we explore this design space. Our ongoing efforts include the tight integration between DuckDB's row groups and the IVF partitioning, morsel-driven parallelism on PDXearch, and dynamically choosing the filtering strategy and partititions to probe based on the predicate's selectivity.

In our ongoing work, we benchmark against the official DuckDB VSS extension and other vector-extended DBMSs (e.g., PostgreSQL with the pgvector extension). As part of future work, we aim to compare our solution to vector systems in general, and explore what PDXearch can offer to lakehouse architectures (e.g., DuckLake).

References

[1] L. Kuffo, E. Krippner, and P. Boncz, "PDX: A Data Layout for Vector Similarity Search," *Proc. ACM Manag. Data*, vol. 3, no. 3, pp. 196:1–196:26, Jun. 2025.

[2] P. Iff, P. Bruegger, M. Chrapek, M. Besta, and T. Hoefler. Benchmarking Filtered Approximate Nearest Neighbor Search Algorithms on Transformer-based Embedding Vectors. [Online]. Available: http://arxiv.org/abs/2507.21989