What is a vector database?

A vector database is a specialized database designed to store and manage data as high-dimensional vectors. The term comes from vectors, which are mathematical representations of features or attributes contained in data. In contrast to traditional databases, which are well suited to handling structured data organized in rows and columns, the vector database structure arranges information as vector representations with a fixed number of dimensions grouped according to their similarity.

Each vector within a vector database consists of a specific number of dimensions, which can vary from just a few dozen to several thousand. The number of dimensions depends on the complexity and granularity of the data. This structure allows vector databases to efficiently handle complex, multifaceted information and perform rapid similarity-based searches and analyses.

When would I use a vector database?

According to the International Data Corporation (IDC), 80% of the new data created worldwide by 2025 will be unstructured data, such as text, images and video. Learning-based models, such as deep neural networks, are increasingly used to manage this unstructured data for applications across industries, from e-commerce to healthcare. These applications work by turning the unstructured data into embedding vectors. Once the data have been "vectorized," tasks such as searches, making recommendations and analysis can be implemented via similarity-based Vector Search. The management of vector data takes place in vector databases.

Knowing when to use vector databases depends on the other processes and technologies you are using. They are a key component to powering many AI systems, and some (but not all) large language model (LLM) applications use vector databases for fast similarity searches or to provide context or domain knowledge. For example, they play a crucial role in retrieval augmented generation (RAG), an approach where the vector database is used to enhance the prompt passed to the LLM by adding additional context alongside the query.

Vector databases also enable hybrid search. This approach combines traditional keyword-based search with semantic similarity search to locate relevant information even when keywords are not an exact match. Vector databases can also be used for a number of natural language processing (NLP) tasks, including semantic and sentiment analysis, or in training machine learning (ML) models.

What is a vector?

A vector is a high-dimensional numerical array that expresses the location of a particular point across several dimensions. Picture a word vector space as a three-dimensional cloud where words are represented as points. In this space, words with related meanings cluster together.

For example, the point representing “apple” would be positioned closer to “pear” than to “car.” This spatial arrangement reflects the semantic relationships between words, with proximity indicating similarity in meaning.

What is vector embedding?

A vector is generated by applying an embedding function to the raw data to transform it into a representation. These representations are called “embeddings” because an ML model takes a representative grouping and embeds it into a vector space. The vectors are embedded as lists of numbers, making it easier for ML models to perform operations with the data. In fact, the performance of ML methods critically depends on the quality of the vector representations. A whole paragraph of text or a group of numbers can be reduced to a vector, allowing the model to perform operations efficiently.

How do vector databases work?

Vector databases are designed to efficiently store, index and query data through high-dimensional vector embeddings. Once a user inputs a query or request into the vector database, it commences the following sequence of processes:

Vectorization: This first step involves generating embeddings from multimodal content, which could include text, images, audio or video. This process captures the semantic relationships in the data. For example, in text data this process ensures that words with similar meanings (or vectors) will be placed close to each other in the vector space.

Vector indexing: The next step sets vector databases apart from traditional databases. ML algorithms, such as product quantization or Hierarchical Navigable Small World (HNSW), are applied to the data to map the vectors to new data structures. These structures enable faster similarity or distance searches, such as nearest neighbor searches between vectors. This indexing process is essential for the database’s performance, as it allows for quick retrieval of similar vectors.

Query execution: In the final stage, the initial query vector is compared against the indexed vectors in the database. The system retrieves the vectors with the strongest relationships, effectively finding the most relevant information based on semantic similarity rather than exact keyword matches.

These processes allow vector databases to perform semantic searches and similarity-based retrievals, making them ideal for applications like recommendation systems, image and video recognition, text analysis, and anomaly detection.

Benefits of vector databases

Vector databases offer a range of benefits:

High speed and performance: Vector databases can rapidly locate similar data using vector distance or similarity metrics, a process that is integral to NLP, computer vision and recommendation systems. Unlike traditional databases, which are limited to exact matches or predefined criteria, vector databases capture semantic and contextual meaning. This optimizes data retrieval by enabling the performance of more nuanced, context-aware searches beyond simple keyword matching.

Scalability: While traditional databases may face challenges with scalability bottlenecks, latency issues or concurrency conflicts when dealing with big data, vector databases are built to handle vast amounts of data. Vector databases enhance scalability by using techniques like sharding, partitioning, caching and replication to distribute the workload and optimize resource utilization across multiple machines or clusters.

Versatility: Whether the data contains images, videos or other multimodal data, vector databases are built to be versatile. Given their ability to handle multiple use cases, ranging from semantic search to conversational AI applications, vector databases can be customized to meet a variety of business requirements.

Cost-effective: Vector databases offer lower costs due to their efficient handling of high-dimensional data. Unlike querying ML models directly, which can be computationally intensive and time-consuming, vector databases use model embeddings to process the dataset more efficiently.

Integration with ML: Vector databases make it easier for ML models to recall previous inputs, allowing ML to power semantic search, classification and recommendation engines. Data can be identified based on similarity metrics instead of exact matches, making it possible for a model to understand the context of the data.

Five vector database use cases

Vector databases are used across industries for a diverse range of applications and use cases. Here are some of the most common vector database examples:

Large language models (LLMs)

The rise of LLMs for tasks like information retrieval, alongside the increased popularity of e-commerce and recommendation platforms, requires vector database management systems that can deliver query optimization capabilities for unstructured data.

In multimodal applications, data is embedded and stored in vector databases, facilitating efficient retrieval of vector representations. When a user submits a text query, the system uses both the LLM and the vector database: the LLM provides NLP capabilities, while the vector database's algorithms perform approximate nearest neighbor searches. This approach can produce better results compared to using either component in isolation.

Vector databases are increasingly being applied to LLMs through RAG, which allows for increased explainability by applying context to LLM outputs. User prompts can be augmented through the inclusion of context to mitigate core LLM challenges, such as hallucination or bias.

Image recognition

Vector databases can play a key role in image recognition by storing high-dimensional embeddings of images generated by ML models. As vector databases are optimized for similarity search tasks, this makes them ideal for applications such as object detection, facial recognition and image search.

Vector databases are fine-tuned for the rapid retrieval of context through similarity. E-commerce platforms can use vector databases to find products with similar visual attributes, while social media sites can suggest related images to users. An illustrative example is Pinterest, where vector databases power content discovery by representing each image as a high-dimensional vector. When a user pins an image of a coastal sunset, the system can swiftly search its vector database to suggest visually similar images, like other beach landscapes or sunsets.

Natural language processing (NLP)

Vector databases have revolutionized NLP by enabling efficient storage and retrieval of distributed word representations. Models like Word2Vec, GloVe and BERT are trained on massive text datasets to generate high-dimensional word embeddings that capture semantic relationships, which are then stored in vector databases for fast access.

As they enable rapid similarity searches, vector databases allow models to find contextually relevant words or phrases. This capability is particularly valuable for tasks like semantic search, question answering, text classification and named entity extraction. Moreover, vector databases can store sentence-level embeddings, capturing word contexts and enabling more nuanced language understanding.

Recommendation systems and personalization

Once a vector database is trained using an embedding model, it can be utilized to generate personalized recommendations. When a user interacts with the system, their behavior and preferences are used to generate the user's embedding. For example, a user can ask an LLM for a TV series recommendation and the vector database can suggest TV series that have plots or ratings similar to the user's preferences. TV series with embeddings closest to the user's encoding are then recommended accordingly.

Fraud detection

Financial institutions use vector databases to detect fraudulent transactions. Vector databases allow companies to compare transaction vectors with known fraud patterns in real time. The scalability of vector databases also allows them to manage risk and acquire new insights into

consumer behavior. These databases can identify patterns that indicate activities by encoding transaction data as vectors. Furthermore, they facilitate the evaluation of creditworthiness and consumer segmentation by analyzing data to improve the decision-making process.

Common challenges of vector databases

Despite their many benefits and use cases, a complete understanding of vector databases needs to include their challenges as well.

New data pipelines

Vector databases require efficient data ingestion pipelines where raw, unprocessed data from various sources can be cleaned, processed and embedded with an ML model before it is stored as vectors in the database.

Databricks Vector Search addresses this challenge by offering a comprehensive solution. It automates vector generation, management and optimization, handling real-time synchronization of source data with corresponding vector indices. The software manages failures, optimizes throughput, and performs automatic batch size tuning and autoscaling without the need for manual intervention.

This approach reduces the need for separate data ingestion pipelines, minimizing “developer toil” and allowing teams to focus on higher-level tasks that directly add business value rather than spending time on building and maintaining complex data preparation processes.

Increased security and governance

Vector databases require additional security, access controls and data governance along with the necessary maintenance and management. Enterprise organizations require strict security and access controls over data so that users cannot access GenAI models that link to confidential data.

Many current vector databases either do not have robust security and access controls in place, or require organizations to build and maintain a separate set of security policies. Databricks Vector Search provides a unified interface that defines data policies to track data lineage automatically without the need for additional tools. This ensures LLMs won’t expose confidential data to users who shouldn’t have access.

High level of technical knowledge

As they offer powerful capabilities for similarity searches and the handling of high-dimensional data, vector databases are essential tools for data scientists working with AI and ML models.

Databricks Vector Search stands out as a serverless vector database that eliminates the need for manual configuration, allowing data scientists to focus on core work rather than infrastructure management.

Key advantages of Databricks Vector Search include seamless integration with lakehouse architecture, automated data ingestion and up to five times faster results compared to other popular vector databases. It is also compatible with existing data governance and security tools through Unity Catalog, ensuring data protection and compliance.

Databricks Vector Search offers flexibility for both novice and advanced users, with automated scaling for data ingestion and querying, as well as plug-and-replace APIs for those who prefer more control over their pipelines. This combination of ease of use and powerful performance simplifies building a vector database for data scientists at all levels of expertise.

Vector databases vs. graph databases

Vector databases organize data as points in a multidimensional vector space. Each point represents a piece of data, and the location reflects its characteristics relative to other pieces of data. This vector database structure is well suited to many GenAI applications, as vector embeddings are generated by LLMs and data can be searched and retrieved easily.

By contrast, graph databases organize data by storing it in a graph structure. Entities are represented as nodes on a graph, while the connections between these data points are represented as edges. The graph structure enables the data items in the store to be a collection of nodes and edges, with the edges representing the relationships between the nodes. The interconnected structure of graph databases makes them well suited for scenarios where the connections between data points are as important as the data itself.

What's the difference between a vector index and a vector database?

A vector index and a vector database serve distinct but complementary roles in handling high-dimensional data.

Vector index: A vector index is a specialized data structure designed to facilitate fast similarity searches among vector embeddings. It significantly enhances search speed by organizing vectors in a way that allows efficient retrieval. Examples of vector indices include Facebook AI Similarity Search (FAISS), HNSW and locality-sensitive hashing (LSH). These indices can be used as stand-alone algorithmic processes or integrated into larger systems to optimize search operations.

Vector database: On the other hand, a vector database is a comprehensive data management solution that not only incorporates vector indexing but also provides additional functionalities

like data storage; create, read, update and delete (CRUD) operations; metadata filtering; and horizontal scaling. It is designed to manage and query vector embeddings efficiently, supporting complex operations and ensuring data integrity and security.

Future trends for vector databases

The recent rise of LLMs and GenAI applications more generally has contributed to a concomitant uptake in vector databases. As AI applications continue to mature, the development of new products and the changing needs of users will decide the direction of future trends in vector databases — however, there are some generally expected directions for this technology.

Increased integration with ML models: The relationship between vector databases and ML models is the subject of increased research. These efforts aim to reduce the size and dimensionality of vectors, minimizing storage requirements for large datasets and boosting computational efficiency.

RAG customization: RAG is an approach used to improve the context provided to an LLM in GenAI use cases, including chatbot and general question-answer applications. The vector database is used to enhance the prompt passed to the LLM by adding extra context alongside the query.

Multi-vector search: Further research is expected on improving multi-vector search capabilities, which is important for applications such as face recognition. Current techniques often rely on combining individual scores, but this approach can be computationally expensive as it increases the number of distance calculations required.

Hybrid search: The evolution of search systems has led to a growing adoption of hybrid approaches that combine traditional keyword-based methods with modern vector retrieval techniques.

How to create a vector database with Databricks

Databricks Mosaic AI Vector Search is Databricks' integrated vector database solution for the Data Intelligence Platform. This fully integrated system eliminates the need for separate data ingestion pipelines and applies security controls and data governance mechanisms, ensuring consistent protection across all data assets.

Databricks Vector Search provides a high-performance, out-of-the-box experience, allowing LLMs to quickly retrieve relevant results with minimal latency. Users benefit from automatic scaling and optimization, removing the need for manual tuning of the database. This integration streamlines the process of storing, managing and querying vector embeddings, making it easier for organizations to implement AI applications, such as recommender systems and semantic searches, while maintaining data security and governance standards.