HUAWEI

From Nearest Neighbor Search to Vector Databases

HKRC

Global Talents, World-class Innovation

汇聚全球英才，引领世界创新

Dr. RUAN Boyu

Huawei HKRC

Outline

- Cutting-edge Technologies: LLM, RAG, and Vector Databases
- Nearest Neighbor Search and Data Dimensionality
- Approximate Nearest Neighbor Search
- Conclusion

HKRC

Large Language Models (LLM)

ChatGPT

Claude

DeepSeek

CURSOR

ChatPDF

Notion AI

Gemini

LLMs understand human input by converting it into **high-dimensional vectors** that represent the meaning, context, and relationships within the text. These vectors enable the model to process and generate coherent, context-aware responses.

HUAWEI

Large Language Models (LLM)

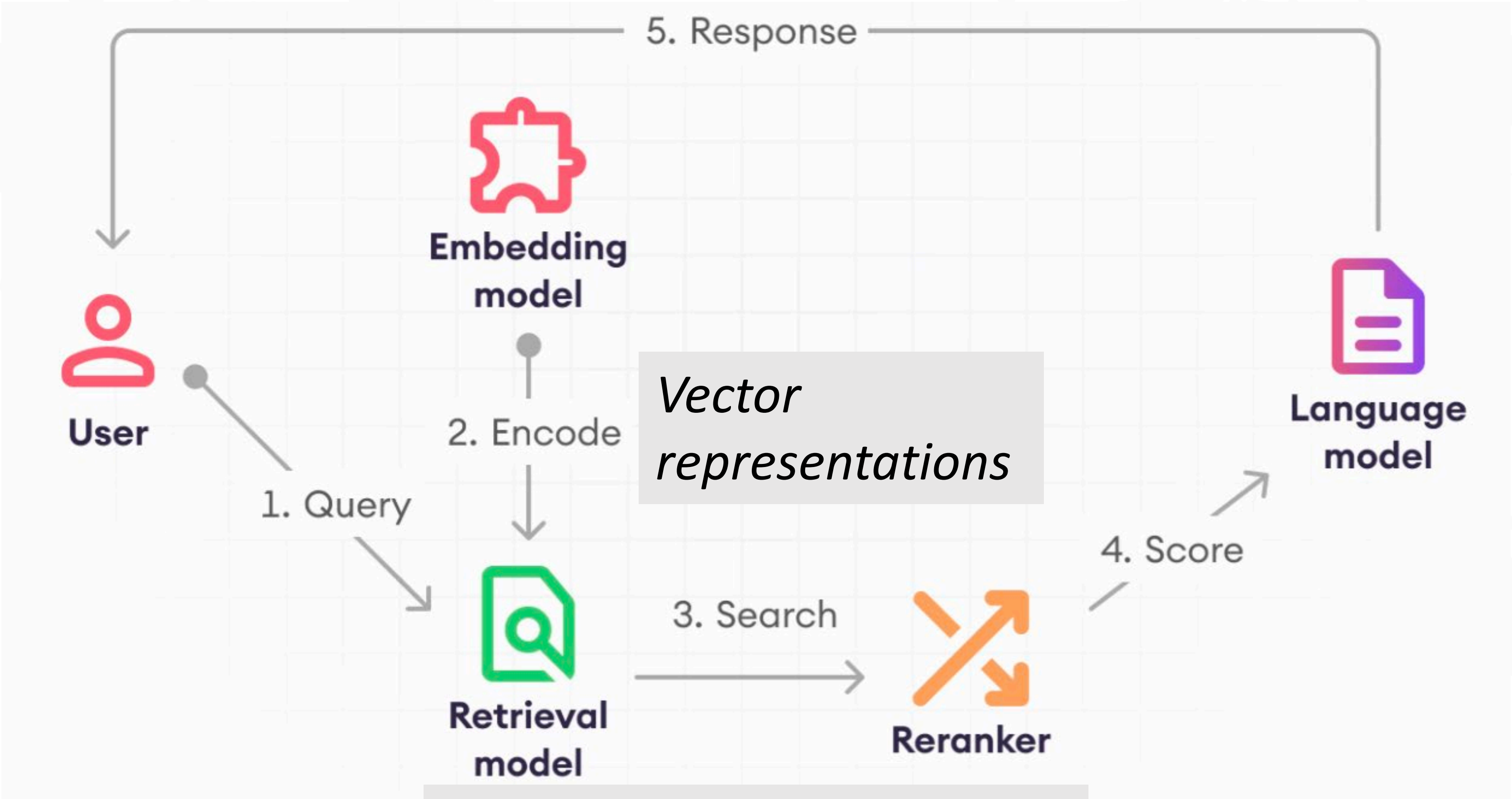
- LLMs are helpful only when it wouldn’t make mistakes
- **Hallucinations**: A hallucination in LLM is a response that contains nonsensical or factually inaccurate text.

FORBES > BUSINESS

BREAKING

Lawyer Used ChatGPT In Court—And Cited Fake Cases. A Judge Is Considering Sanctions

LLMs, RAG, and Vector DB Work Together

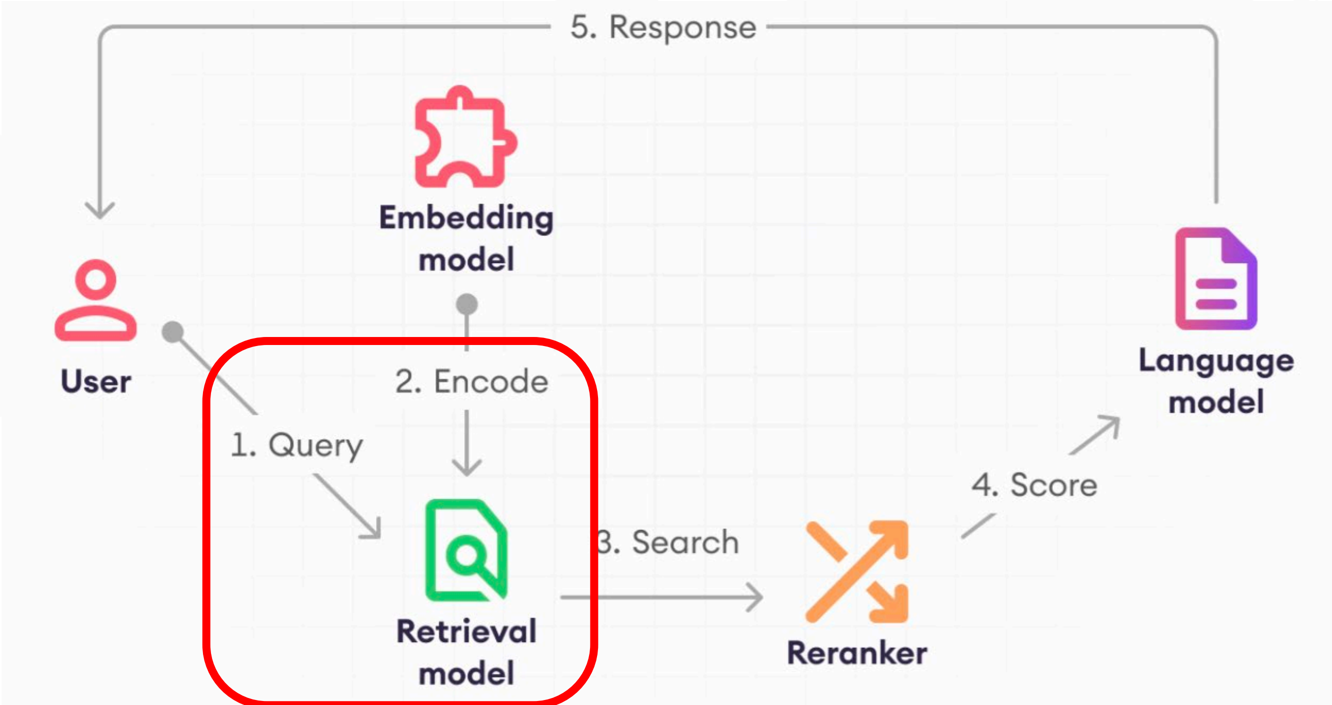


Vector representations

external knowledge from Vector Databases

Retrieval Augmented Generation (RAG)

- Bridge the gap by augmenting LLMs with external knowledge.
  - **Query**: A question that a user asks → encoded as vectors
  - **Augmentation**: Retrieve relevant contents from knowledge bases
  - **RAG**: Use these as context and let LLMs generate better answer.

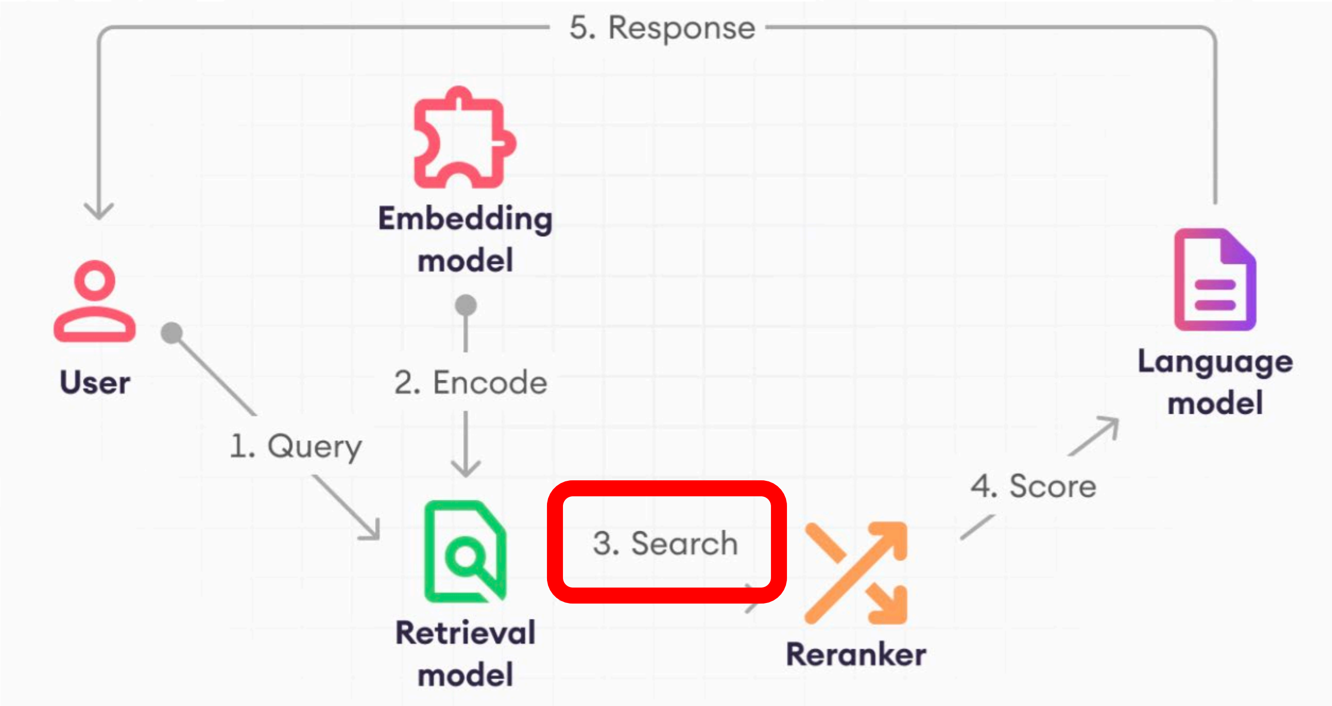


Retrieval Augmented Generation (RAG)

- **Use Cases**
  - ChatPDF: PDF files as knowledge bases
  - Personalized recommendation: purchase history as knowledge bases
  - GraphRAG: Knowledge graphs as knowledge bases

- **Why RAG?**
  - When querying personal or non-public information
  - When querying information in a specific domain
  - When querying changing or dynamic information

Retrieval Augmented Generation (RAG)



- **Encode (Embed):** Transform knowledge and queries into vectors
- Objective: To ensure similar knowledge have similar embedding vectors

Retrieval Augmented Generation (RAG)

- How to store diverse knowledge bases?
  - It would be extremely costly if transforming knowledge bases each time.
  - We need to find a way to store and manage vectors.

- Relational Data – Relational Databases (MySQL, PostgreSQL)
- Graph Data – Graph Databases (Neo4j)
- Vector Data - ?

PostgreSQL

MySQL

HKRC

Vector Databases

- A specialized database designed to store, index, and query data represented as high-dimensional vectors.
- **Existing Vector DBs**
  - PGVector: a open-source PostgreSQL extension designed for vector embeddings directly in a PostgreSQL database.
  - Milvus: A purpose-built distributed vector database optimized for large-scale, high-performance vector search (millions or billions of vectors)
  - FAISS: A library for high-performance vector similarity search and clustering on a single machine (vectors are stored in memory)
  - GaussVector (developed by Huawei)
  - …

How to search relevant knowledge in vector DBs?

HUAWEI

HKRC

Nearest Neighbor Search and Data Dimensionality

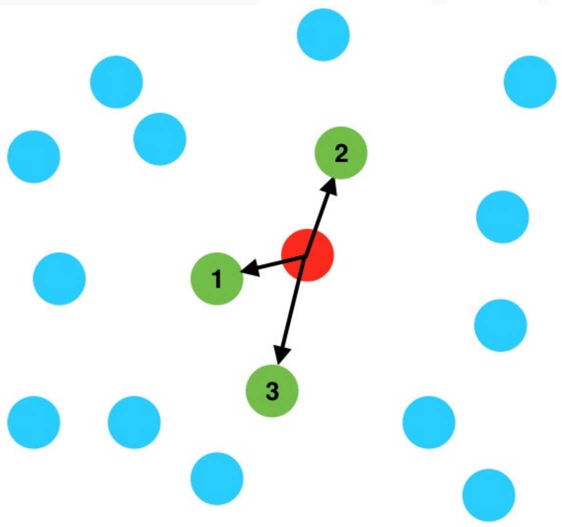
Binary search, B-tree, kd-tree, …

Nearest Neighbor Search (NNS)

- The idea of nearest neighbor search was first formally defined in 1951

- **NN Search**: the process of finding the closest data points (neighbors) to a given query point in a dataset, based on a chosen distance or similarity metric (e.g., Euclidean distance, cosine similarity).

- We often use the concept of *k*-NN.

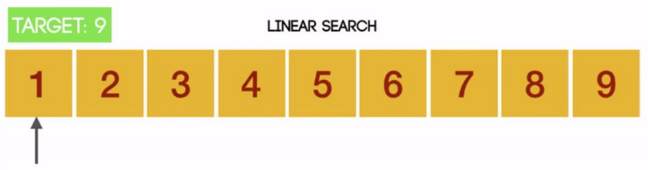


Nearest Neighbor Search (NNS)

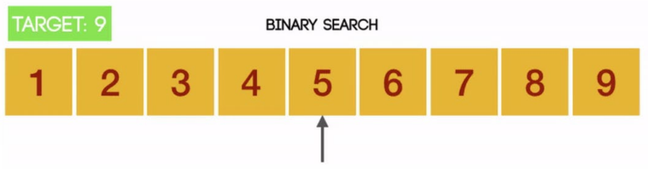
- Many real-life applications involve NNs.
  - **Recommendation in E-commerce**: suggest products to users based on their browsing or purchasing history
  - **Streaming Platforms**: recommend movies, songs, or shows by finding content similar to what a user has previously liked (e.g., Spotify, Neflix)
  - **Ride-sharing**: match nearest drives to passengers
  - **Navigation**: recommend nearest facilities (e.g., restaurant)
  - ...

Nearest Neighbor Search (NNS)

- Given a query and a dataset with N points in total, how to find its nearest neighbor?
  - **A simple solution**: calculate the distance from the query point to all N points in dataset → N times distance calculation



- A simple improvement: Binary Search
  - Sort the N numbers from small to large
  - Iteratively compare the query q with the number in the middle mid
    - If $q = mid$, we find NN (distance=0) and exit.
    - If $q < mid$, we discard all numbers to the right (larger than mid)
    - If $q > mid$, we discard all numbers to the left (smaller than mid)



Data Dimensionality

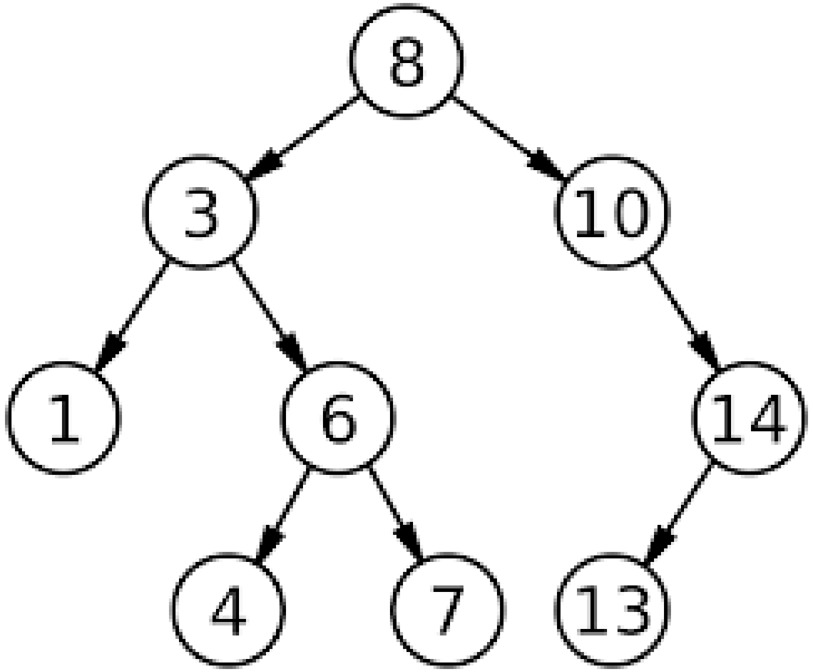
- An object is d-dimensional → it must be described by d attributes.
  - 1-dimensional: age, height
  - 2-dimensional: (latitude, longitude), (temperature, humidity)
  - 3-dimensional: RGB values

- Eculian Distance (L_2): $d(X,Y) = \sqrt{\sum(x_i - y_i)^2}$
- Manhattan Distance (L_1): $d(X,Y) = \sum |x_i - y_i|$
- Chebyshev Distance (L_∞): $d(X,Y) = \max |x_i - y_i|$
- Inner Product Similarity: $s(X,Y) = X\ Y = \sum x_i y_i$
- Cosine Distance: $d(X,Y) = 1 - \frac{X\ Y}{||X|| \cdot ||Y||}$

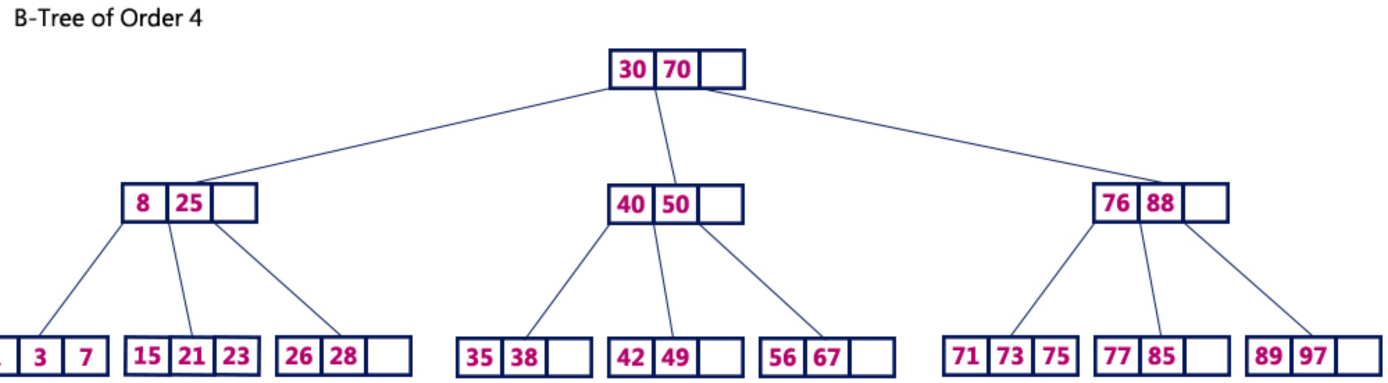
Cost: calculating distances requires d times calculation

Index for NN Search

- **Binary Search Tree**



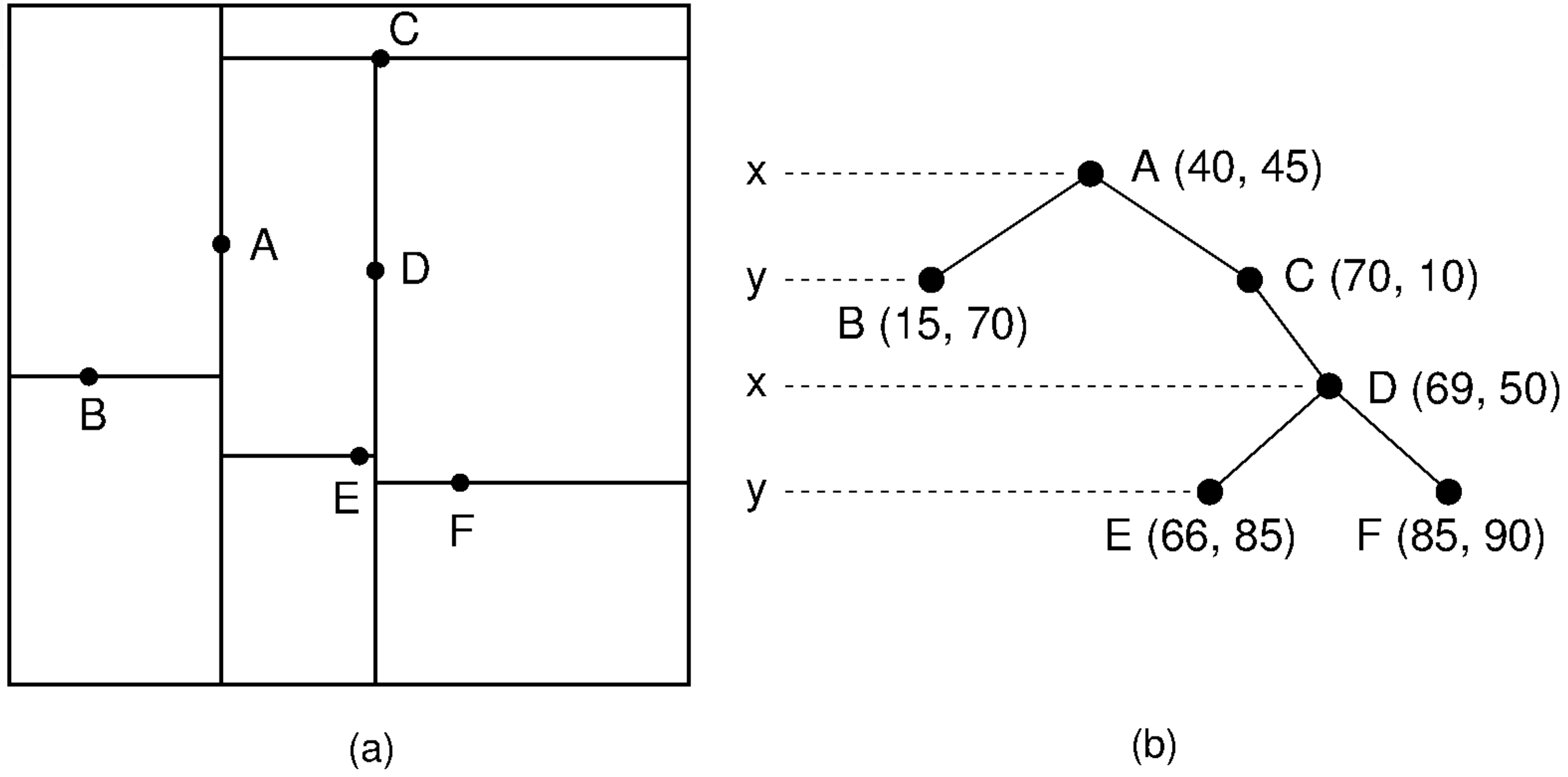
- **B-Tree**



Bayer, Rudolf, and Edward McCreight. "Organization and maintenance of large ordered indices." Proceedings of the 1970 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control. 1970.

Index for NN Search

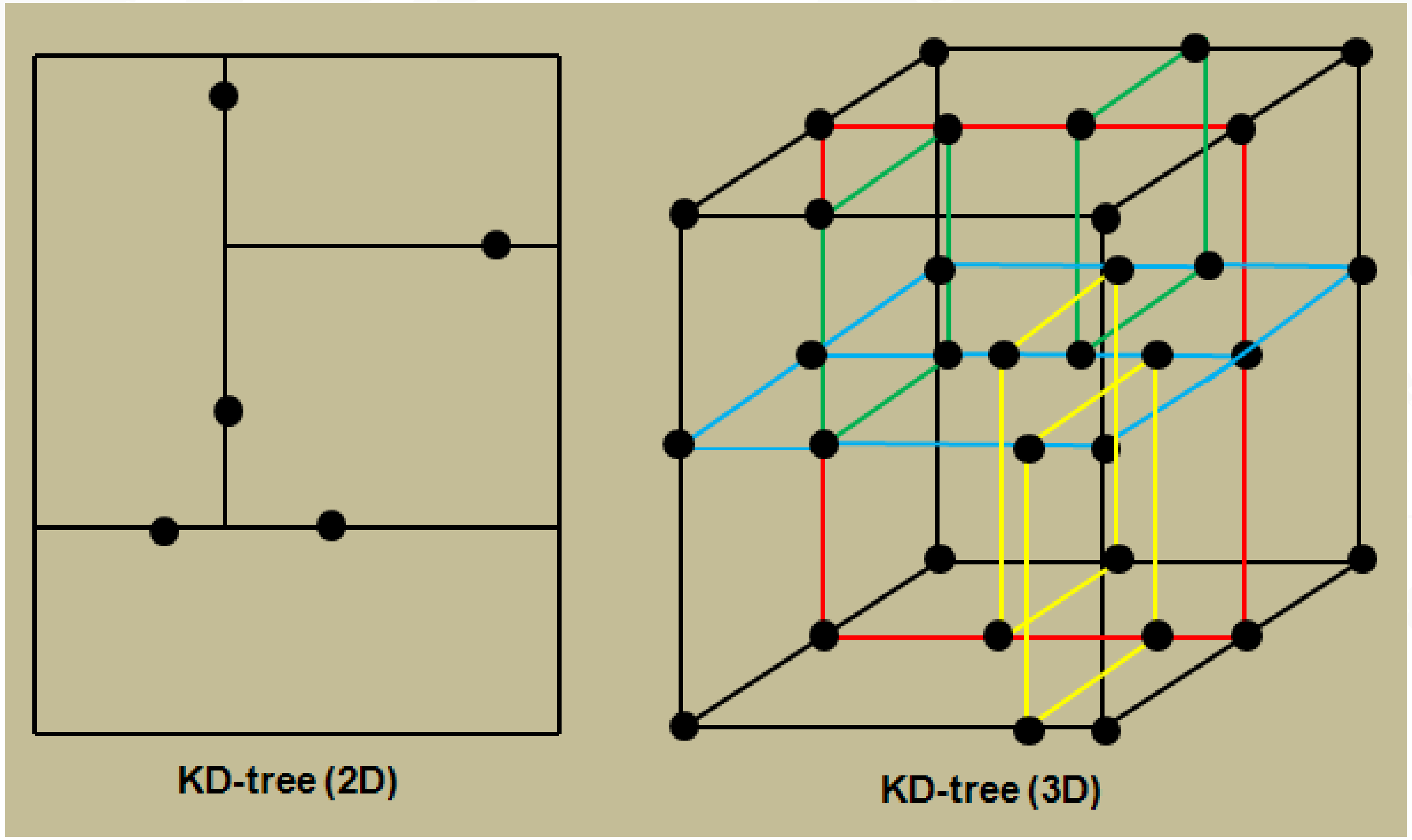
- **K-d tree**



Bentley, Jon Louis. "Multidimensional binary search trees used for associative searching." *Communications of the ACM* 18.9 (1975): 509-517.

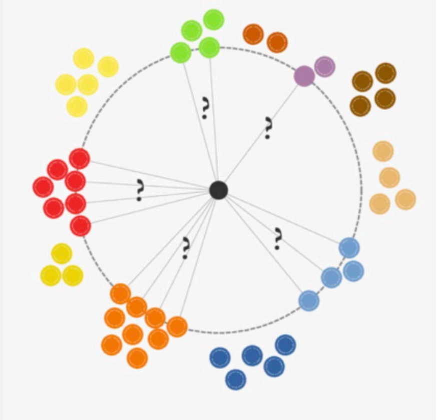
Index for NN Search

- K-d tree



Bentley, Jon Louis. "Multidimensional binary search trees used for associative searching." *Communications of the ACM* 18.9 (1975): 509-517.

As the dimensionality goes higher …



- The curse of dimensionality
  - The size of index goes higher, and efficiency goes lower
  - The meaningfulness of “nearest” is low in high-dimensional space
- Modern embedding models transform knowledge into vectors with 1000+ dimensions
  - **OpenAI**: text-embedding-3-small - 1536 dimensions
  - **Nvidia**: NV-Embed-V2 – 4096 dimensions
- Finding “exact” nearest neighbor is infeasible and meaningless!

Approximate Nearest Neighbor Search

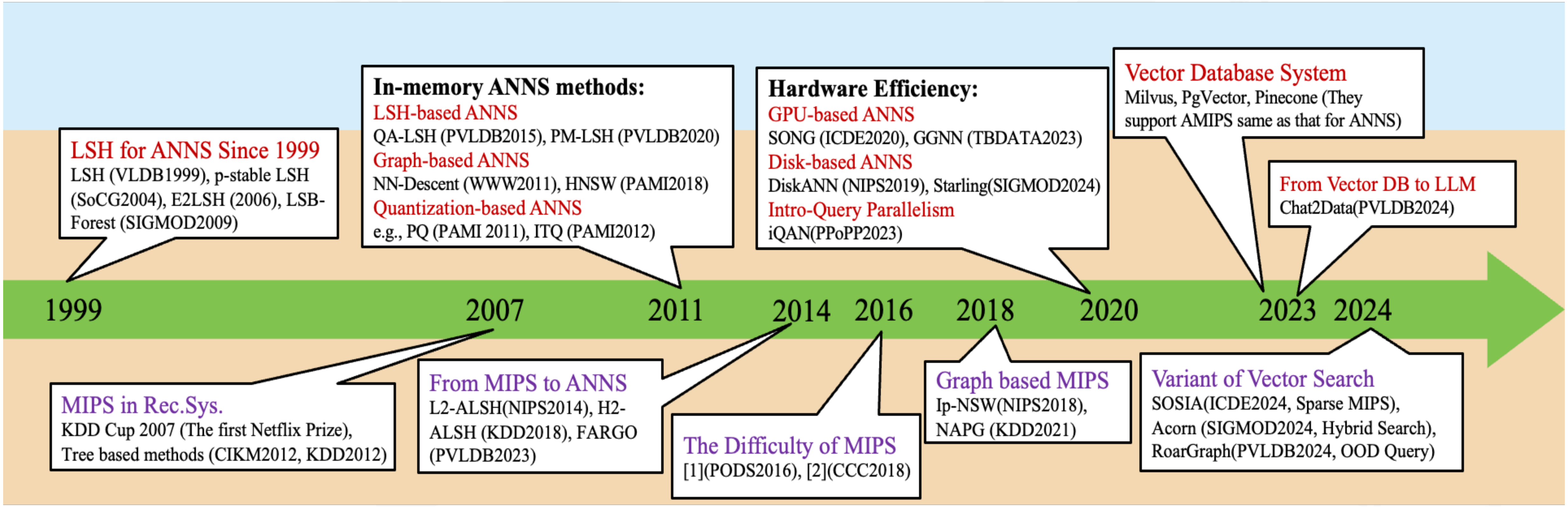
LSH-based, Quantization-based, and Graph-based methods

Approximate Nearest Neighbor Search (ANNS)

- **Facts**
  - High-dimensional data is inevitable, with increasing dimensionality
  - Similarity queries in high-dimensional spaces are also inevitable
  - Queries must meet two conditions: efficiency and meaningfulness

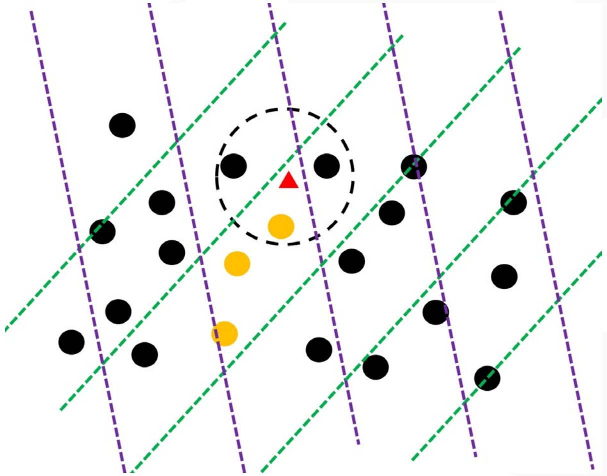
- **An intuitive idea: dimension reduction** (e.g., PCA, LDA, etc)
  - Useful in some cases
  - Information loss
  - The dimension is still high after reduction

Approximate NN Search in High-Dimension

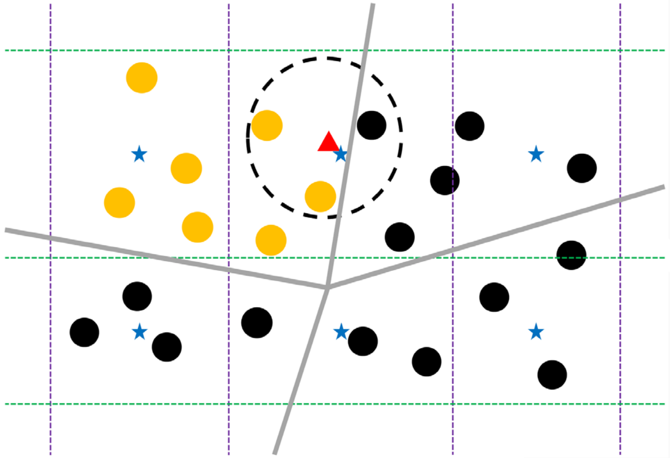


Zhou, Xiaofang. "Efficient Query Processing in Vector Databases“, Keynote, ADC 2024.

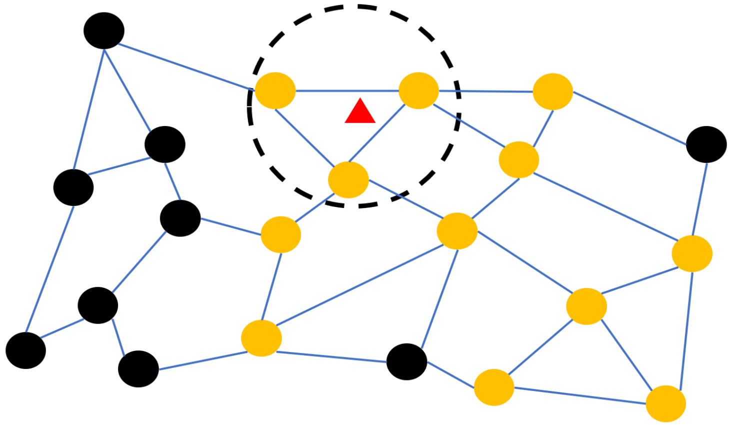
Approximate NN Search in High-Dimension



LSH-based methods



Quantization-based methods

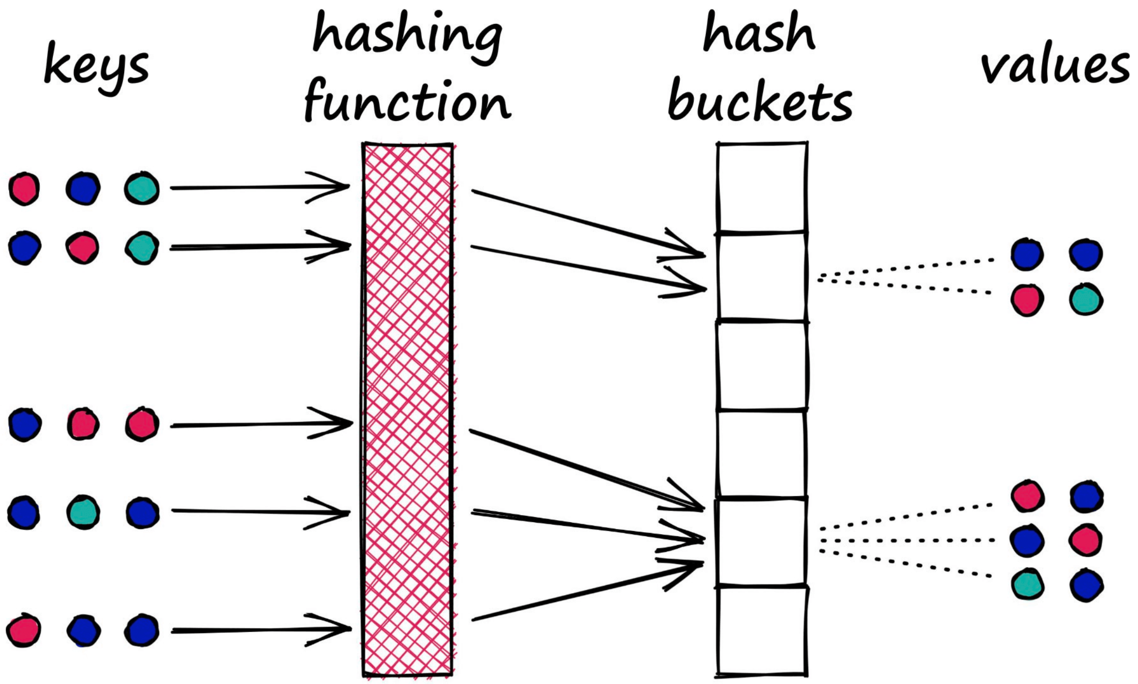


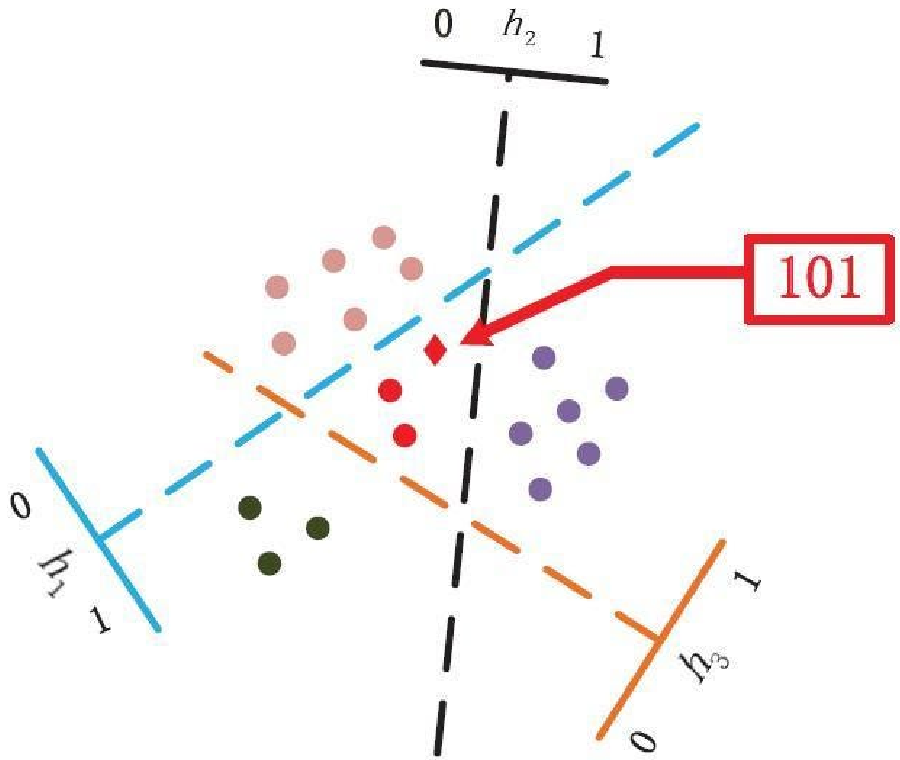
Graph-based methods

Zhou, Xiaofang. "Efficient Query Processing in Vector Databases“, Keynote, ADC 2024.

LSH-based Methods

- LSH (locality-sensitive hashing)

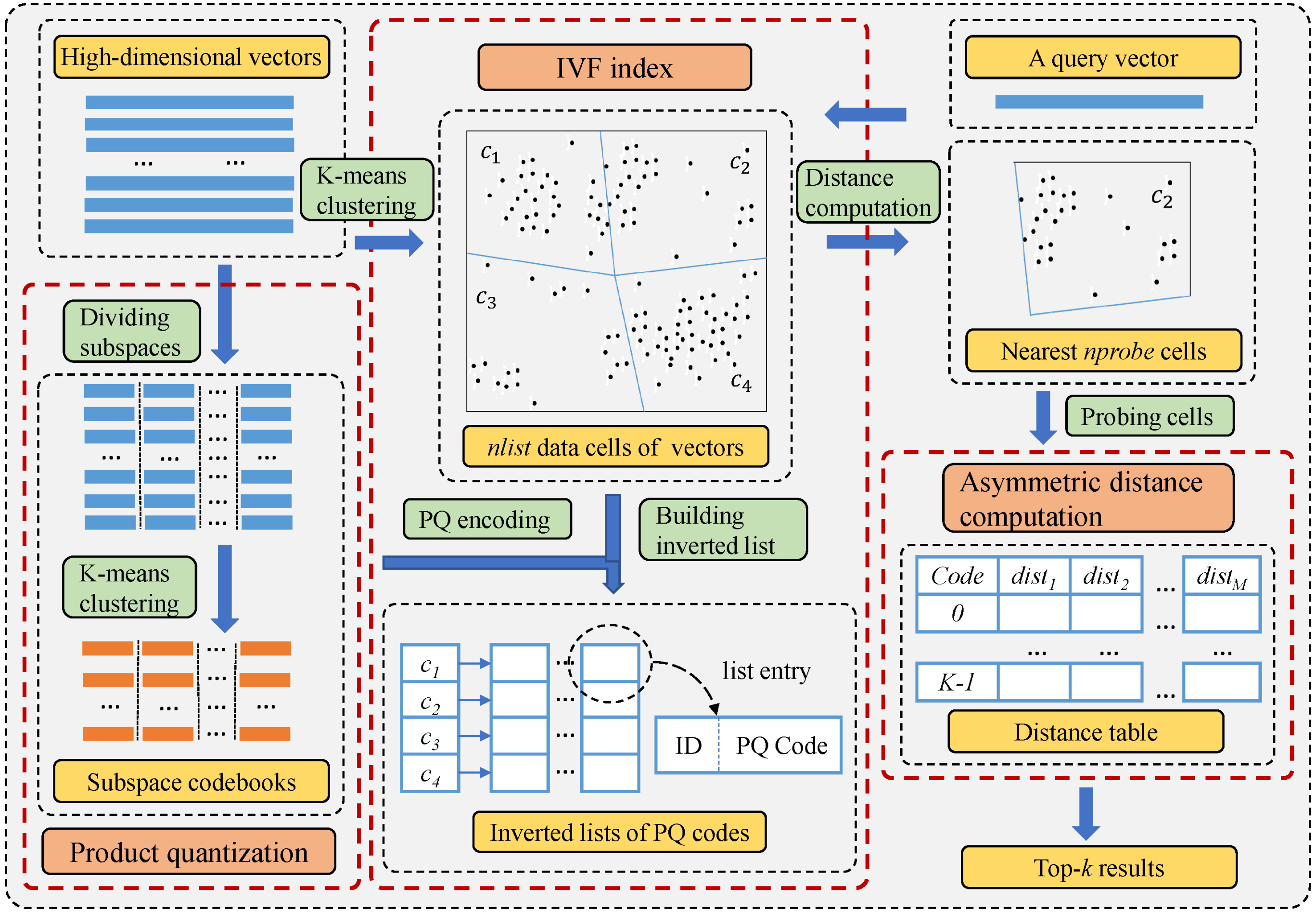




Datar, Mayur, et al. "Locality-sensitive hashing scheme based on p-stable distributions." Proceedings of the twentieth annual symposium on Computational geometry. 2004.

Quantization-based Methods

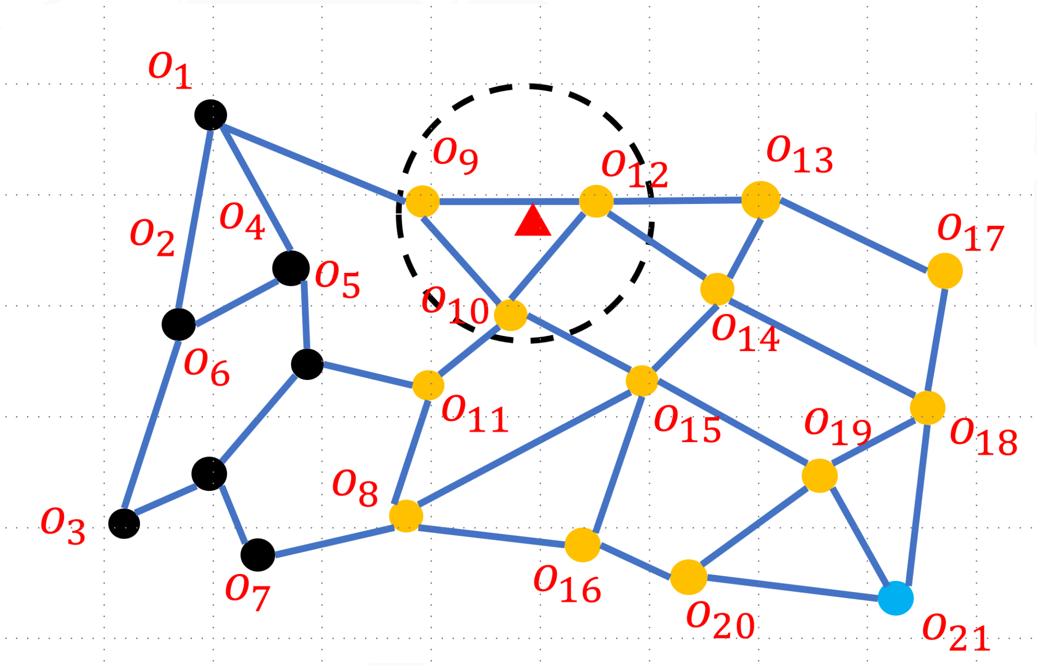
- IVF-PQ (Product Quantization)
  - Product Quantization (PQ)
    - Subspace division
    - Vector quantization in subspaces
    - Subspace code concatenation
  - Inverted File Index (IVF)
    - nlist cells
  - Asymmetric distance measure
    - $dist(q, g(o)) = \sqrt{\sum_{i=1}^{M} dist(q^i, g(o)^i)^2}$

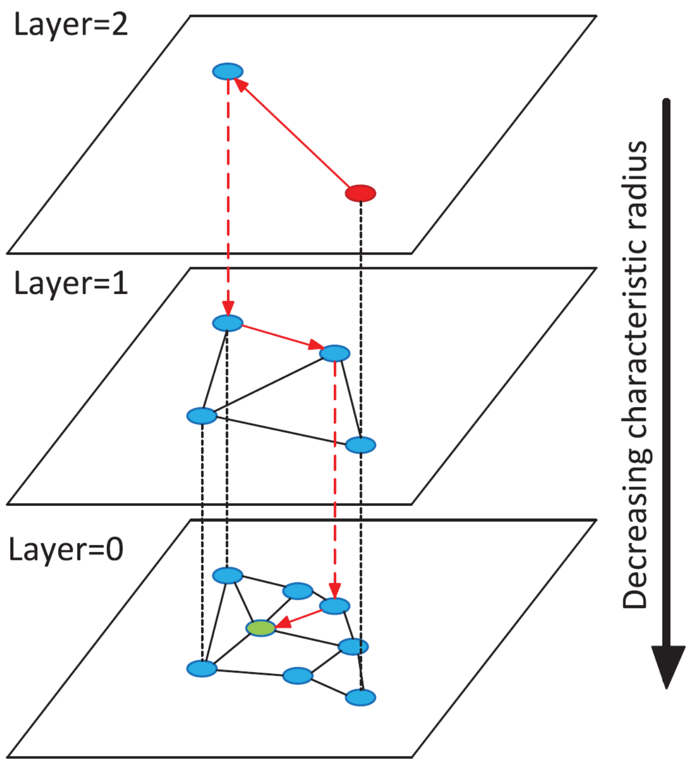


Jegou, Herve, Matthijs Douze, and Cordelia Schmid. "Product quantization for nearest neighbor search." IEEE TPAMI 33.1 (2010): 117-128.

Graph-based Methods

- HNSW (Hierarchical Navigable Small World Graph)





Malkov, Yu A., and Dmitry A. Yashunin. "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs." IEEE TPAMI, 42.4 (2018): 824-836.

Beyond RAG?

- Introduce LLM to end-to-end data management – Data + AI
- Query
  - Intention inference
  - Query decomposition
- Retrieval
  - Hybrid retrieval
  - MoE structure
- Analysis
  - Intelligent agents

Thanks!

Dr. RUAN Boyu
Email: ruan.boyu@huawei.com

