大模型系列 – 向量数据库

相似性搜索概述

大模型业务全流程

数据 & 模型算法

2. 数据处理

- 开源数据
- 数据预处理
- 向量数据库

3. 模型算法

- LLM模型架构
- 多模态一切皆Tokens

模型训练 & 微调

4. 模型训练

- 混合精度
- 梯度检查
- 梯度累积
- …
- 5. 分布式并行
- 训练集群稳定性

6. 模型微调

- 全参微调
- 低参微调
- 指令微调

模型验证 & 推理部署

7. 模型验证

- 下游任务
- 测评标准
- bench mark

8. 推理与智能体

- 量化压缩
- 推理加速
- 9. Agent 智能体

集群算力准备

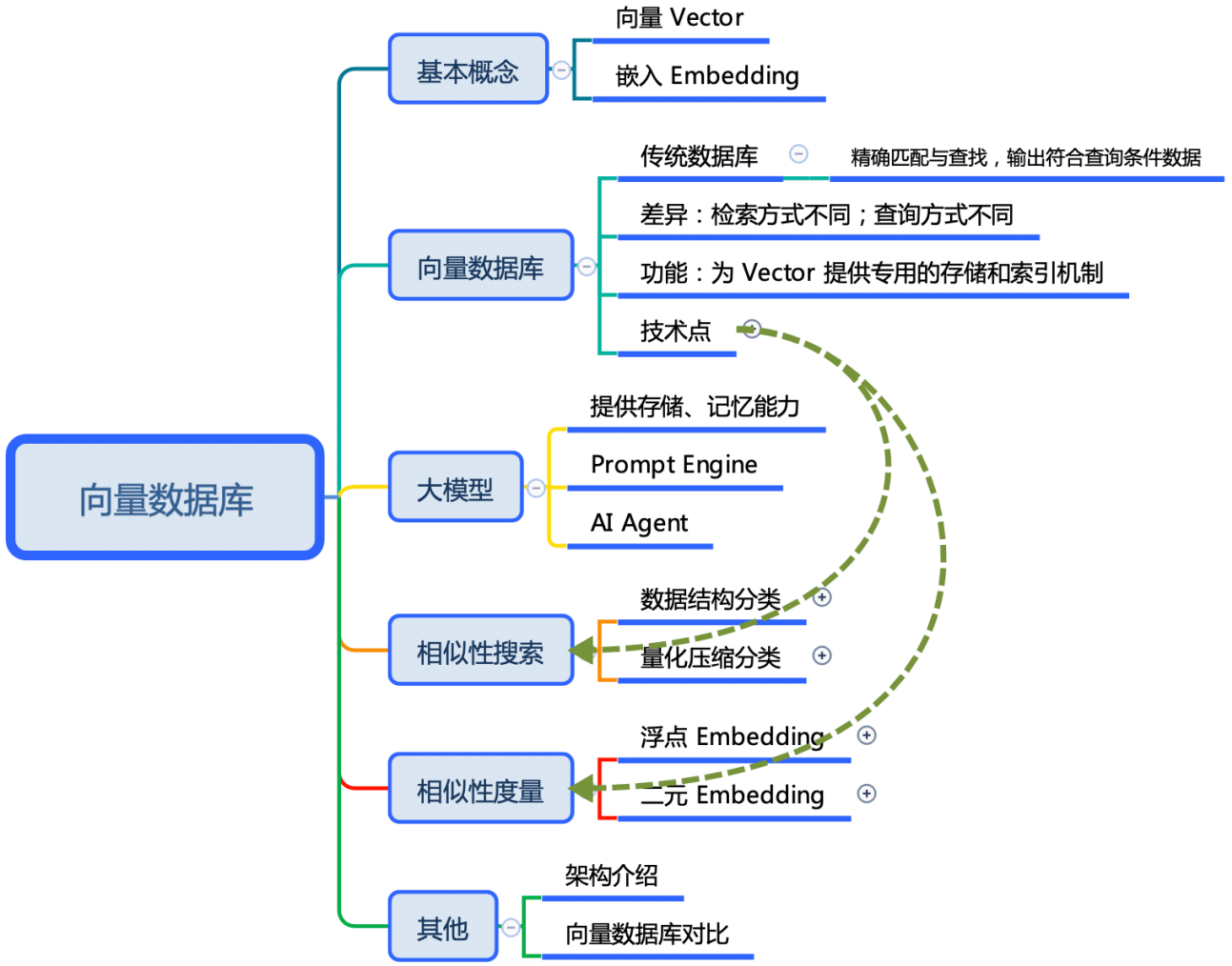
1. AI 集群建设

- 计算、存储、网络
- AI 集群机房建设
- AI 集群上线与运维

大模型系列 – 数据处理之向量数据库

- 具体内容
  - **向量与检索：**向量 Vector 的表示 -- Embedding 原理
  - **向量数据库：**向量数据库原理、功能、特点 -- Vector-DB 应用场景
  - **大模型关系：**向量数据库遇到大模型 – 大模型与 Vector-DB 应用场景
  - **相似性搜索：**K-Means 聚类 -- Faiss 算法 -- PQ 算法 -- IVF 算法 -- HNSW 算法
  - **相似性度量：**欧氏距离（L2） -- 内积（IP） -- 其他度量方式
  - **通用性架构：**通用 Vector-DB 架构 -- KDB 架构示例
  - **对比与小结：**业界向量数据库横向对比 -- Vector-DB 小结

大模型系列 – 数据处理之向量数据库

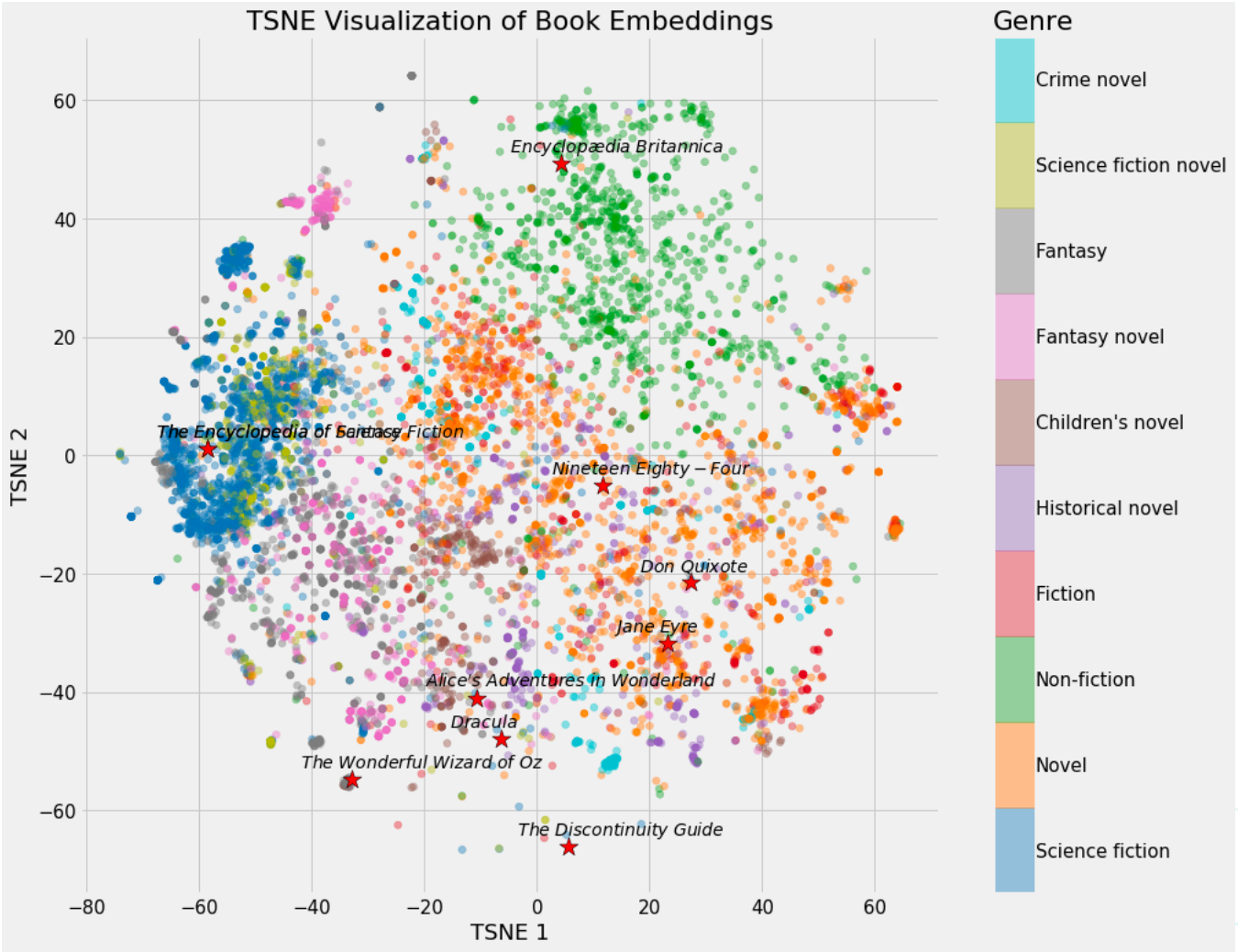


Pinecone	Proprietary composite index
milvus / zilliz	Flat, Annoy, IVF, HNSW/RHNSW (Flat/PQ), DiskANN
Weaviate	Customized HNSW, HNSW (PQ), DiskANN (in progress...)
qdrant	Customized HNSW
chroma	HNSW
LanceDB	IVF (PQ), DiskANN (in progress...)
vespa	HNSW + BM25 hybrid
Vald	NGT
elasticsearch	Flat (brute force), HNSW
redis	Flat (brute force), HNSW
pgvector	IVF (Flat), IVF (PQ) in progress...

向量高维表示

- Embedding 维度足够多，理论上可以将所有 Vector 区分开来；

- 即高维特征空间中对应一个点，世间万物都可以用 Vector 坐标表示；



相似性度量

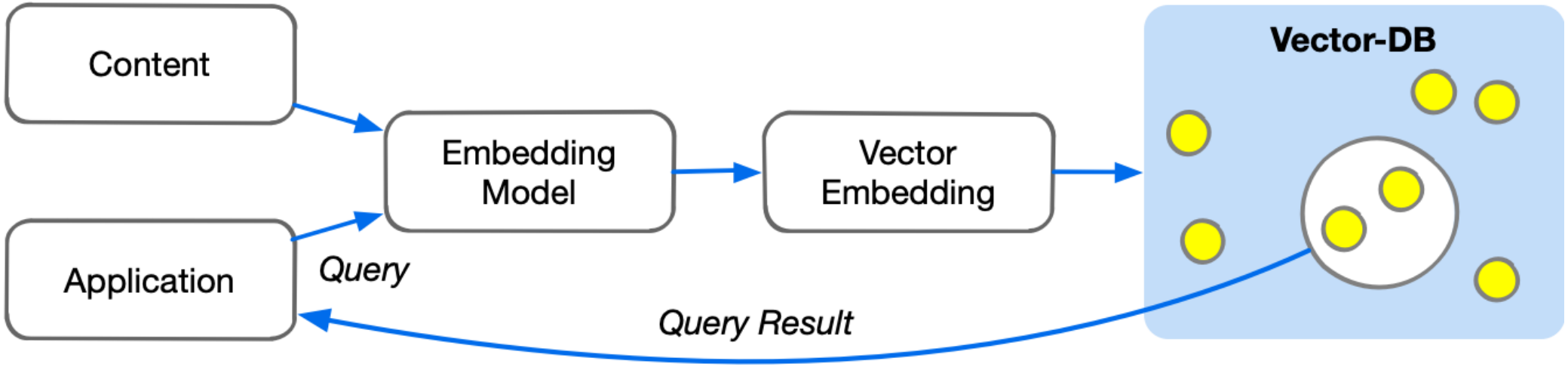
1. 向量可以高维空间表示，和相似性搜索什么关系？
   - e.g.，银渐层和布偶都是猫咪特征接近，将特征用向量来表示（Embedding），Vector 具有大小和方向数学结构，通过计算 Vector 间距离来判断其相似度，即**相似性度量**（Similarity Metrix）。

相似性搜索

1. 可以通过比较 Vector 间距离来判断相似度，如何应用到更大的场景？
   - e.g.，海量向量 $\boldsymbol{A}$ 中找到与某向量 $\vec{a}$ 最相似的向量，需要对 Vector-DB 中每个向量进行计算；
   - q.u.，逐向量对比计算量巨大，所以需要一种高效算法，即**相似性搜索**（Similarity Search）。

Vector-DB 与相似性搜索

- **向量数据库为向量数据提供专门的存储和索引机制。**
  - 向量被存储为高维空间中的点，DB 会为这些点建立索引
  - 索引结构和算法有树、图、哈希等，索引结构让 DB 可高效进行相似度搜索
  - 每次搜索通过相似性度量评价相似度，从而得到向量查询结果

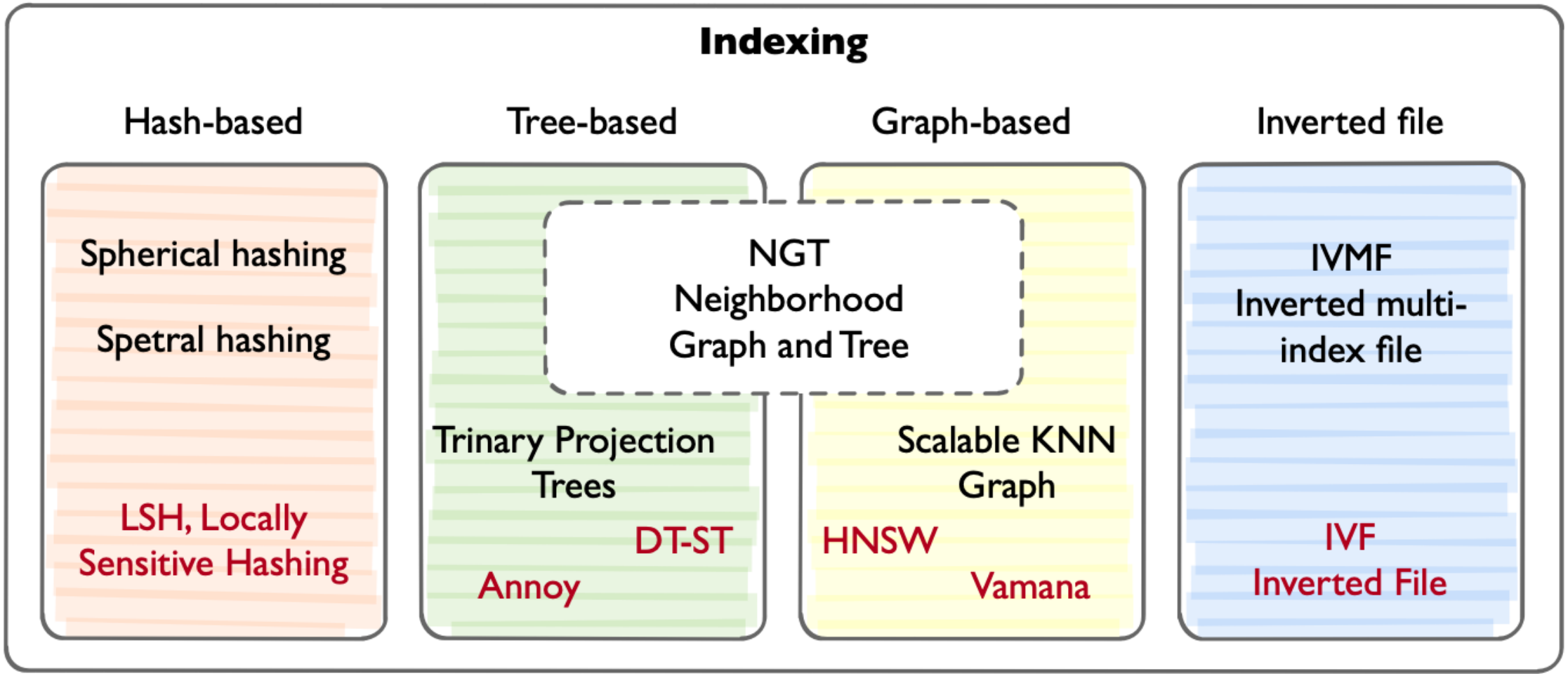


相似性搜索分类

- 高效的搜索算法有很多，其主要思想是通过两种方式提高搜索效率：
  1. **按索引使用数据结构：**将向量组织成基于树、图等结构来缩小搜索范围。
  2. **减少向量大小：**通过降维的方式（量化）表示向量值的长度。

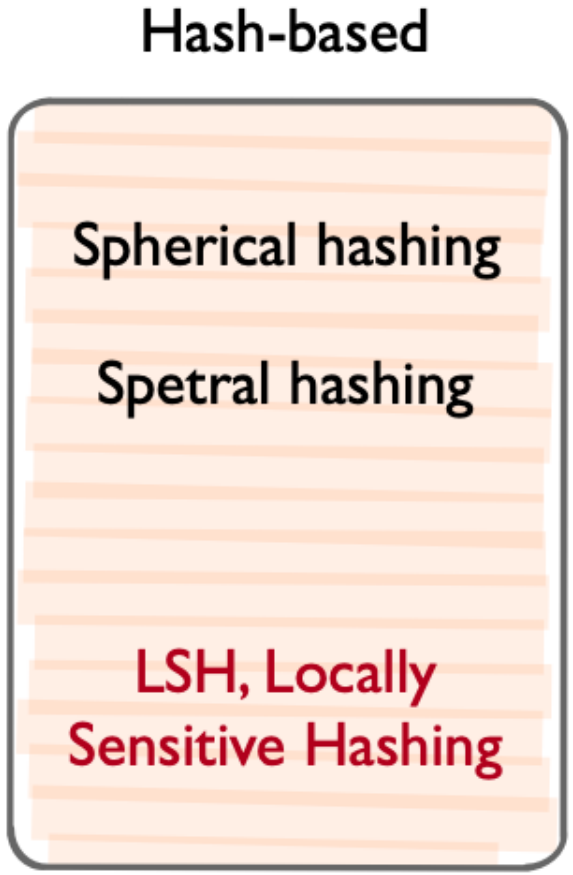
1. 按数据结构分类

按数据结构分类


Indexing
Hash-based
Tree-based
Graph-based
Inverted file
Spherical hashing
Spetral hashing
LSH, Locally Sensitive Hashing
NGT Neighborhood Graph and Tree
Trinary Projection Trees
DT-ST
Annoy
Scalable KNN Graph
HNSW
Vamana
IVMF Inverted multi-index file
IVF Inverted File

基于哈希索引

1. **高维向量映射到低维空间或低维哈希码：**尽可能保持原始相似性；
2. **数据库中向量被多次哈希：**以确保相似点更有可能发生冲突（与传统哈希相反，其目标最大限度减少冲突）；
3. **通过哈希表或者倒排索引来存储和检索：**在索引过程，查询点也使用与索引过程中相同哈希函数，由于相似点被分配到相同哈希桶，因此检索速度非常快；

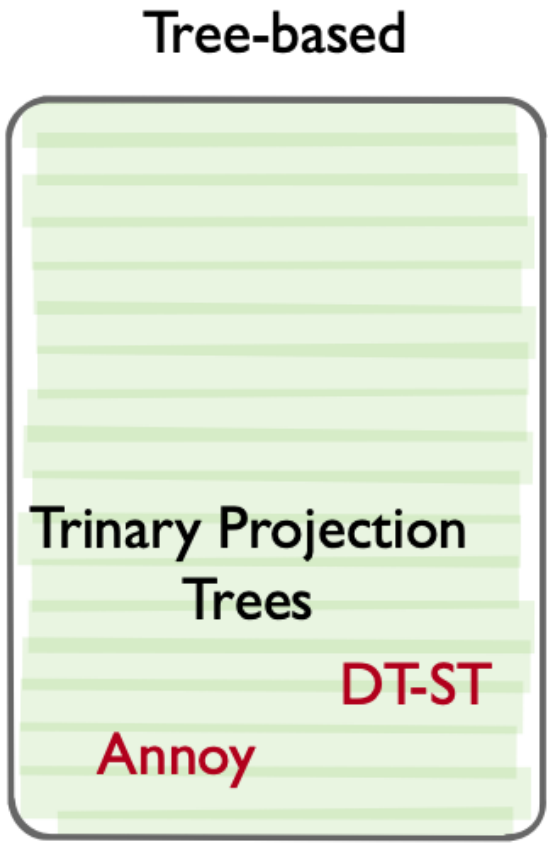


- **典型算法：** LSH 局部敏感哈希
- **优缺点：**优点扩展到大量数据时速度非常快，缺点是准确性一般。

基于树的索引

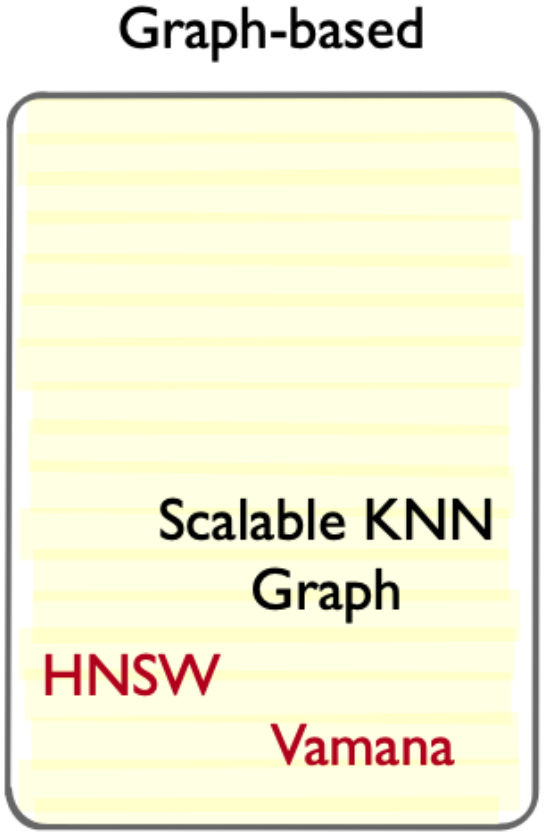
- **建立树结构**：把高维空间划分成若干个子空间或者聚类中心，然后用树形结构来存储和检索。
- **索引算法**：通过二叉搜索树算法搜索，相似数据易在同一子树，从而更快地发现近似邻居。

- **特点**：基于精确距离计算 or 近似距离计算。
- **优缺点**：优点对低维数据准确率较高；缺点无法充分捕获数据复杂性，高维数据准确率较低。



基于图的索引

- **数据结构**：图中节点表示向量数据，边表示数据间相似性。
- **构图方式**：相似数据点更有可能通过边连接，搜索算法以有效方式遍历图找到相似近邻。
- **优缺点**：优点能够在高维数据中找到近似的近邻，从而提高搜索性能。缺点是构图方式复杂，影响内存效率。

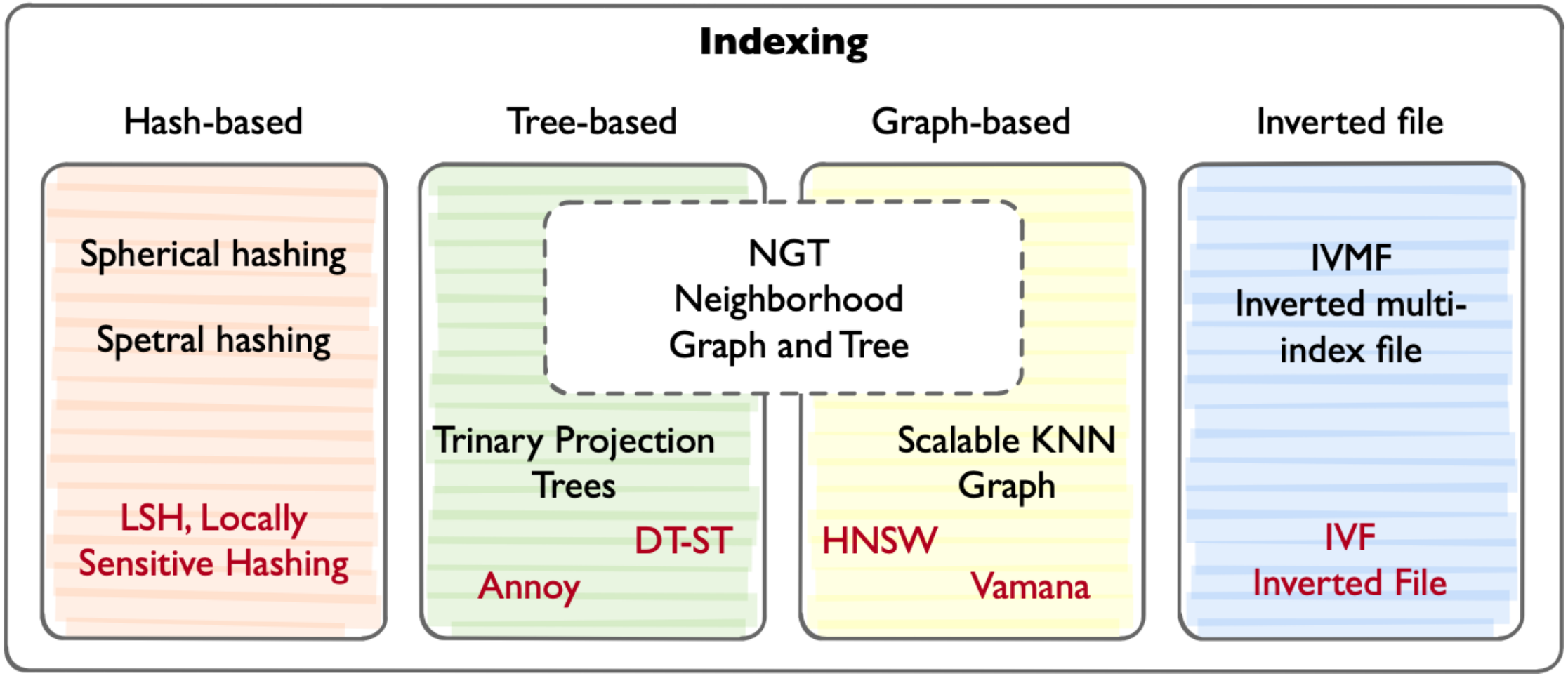


倒排文件索引

- **倒排文件索引（IVF）**：将向量空间划分为多格 Voronoi 单元，单元以与聚类相同的方式，通过建立倒排索引表，以减少搜索空间。

- **优缺点**：优点是有助于设计快速缩小感兴趣相似区域的搜索算法；缺点是对于海量数据，细分向量空间会变慢。
- **改进点**：IVF 常与乘积量化（PQ）等量化方法结合，以提高性能。

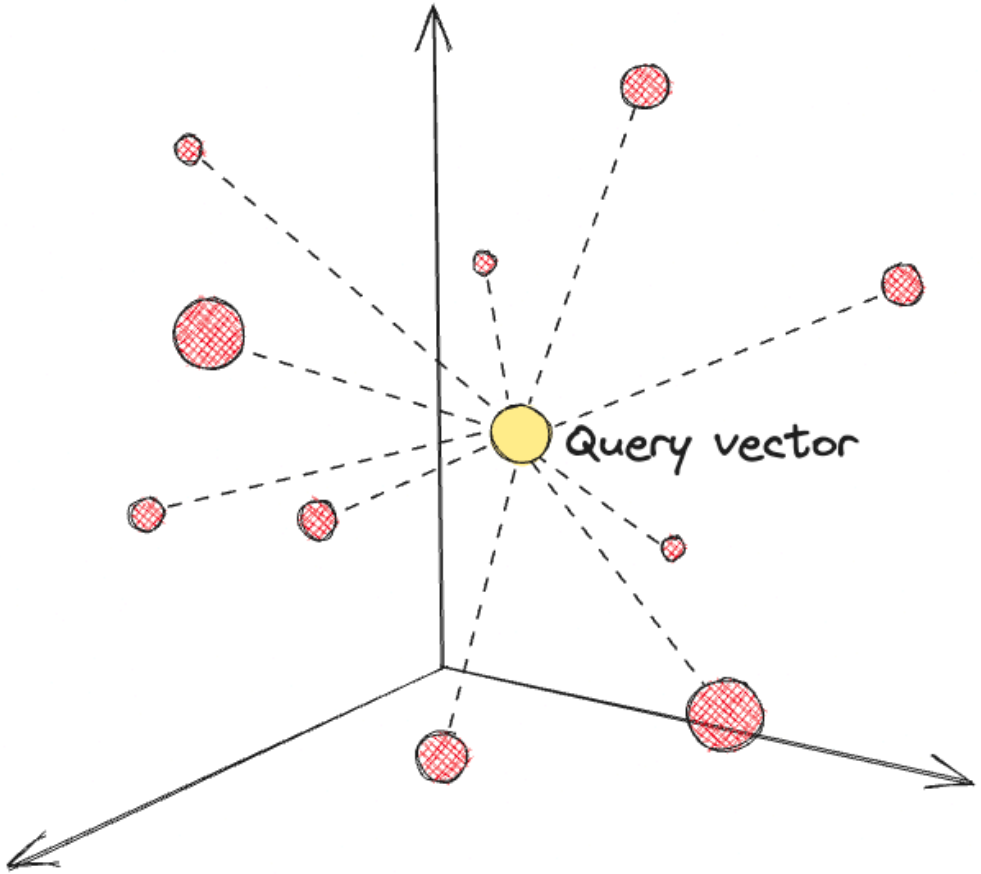
按数据结构分类


Indexing
Hash-based
Tree-based
Graph-based
Inverted file
Spherical hashing
Spetral hashing
LSH, Locally Sensitive Hashing
NGT Neighborhood Graph and Tree
Trinary Projection Trees
DT-ST
Annoy
Scalable KNN Graph
HNSW
Vamana
IVMF Inverted multi-index file
IVF Inverted File

2. 按量化压缩分类

索引的本质

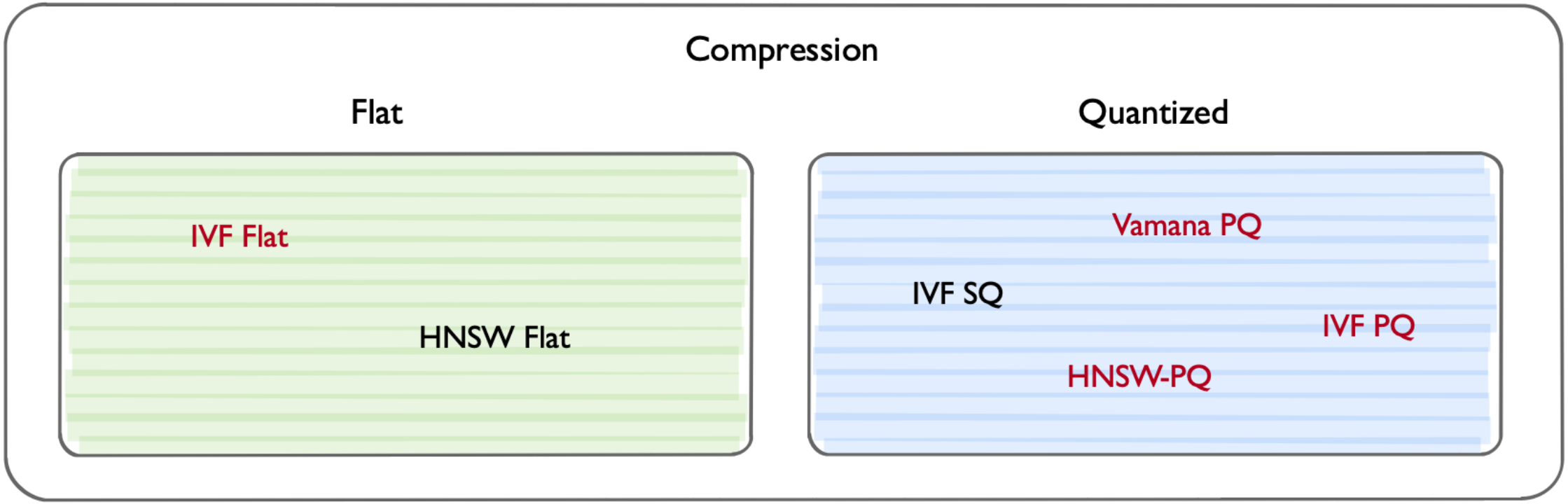
- **常见索引**：对直接存储的向量，使用特殊数据结构设计的算法加快索引效率。
- **具体算法**：查询向量时，与数据库中每个向量进行的比较，e.g. 示例所示。
- **索引本质**：减少了时间搜索效率，返回 TOP-K 最近邻向量。每一次相似性对比所需时间随数据维度增加而增加。
- **减少向量大小**：通过降维的方式（量化）表示向量值的长度。



压缩分类

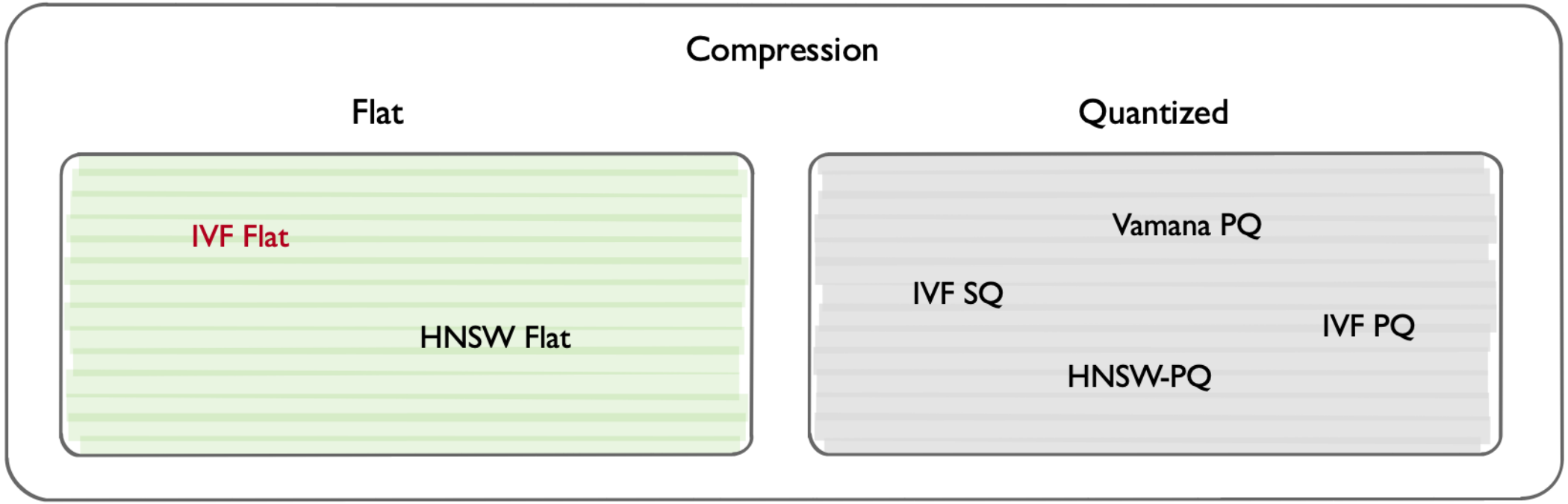
1. **实现的方式？**
   - 索引基础向量被分解为较少字节组成的块，以减少搜索期间的内存消耗和计算成本。
2. **代价是什么？**
   - 通过压缩提高索引效率，代价是降低检索准确性。

压缩分类


Compression
Flat
Quantized
IVF Flat
HNSW Flat
Vamana PQ
IVF SQ
IVF PQ
HNSW-PQ

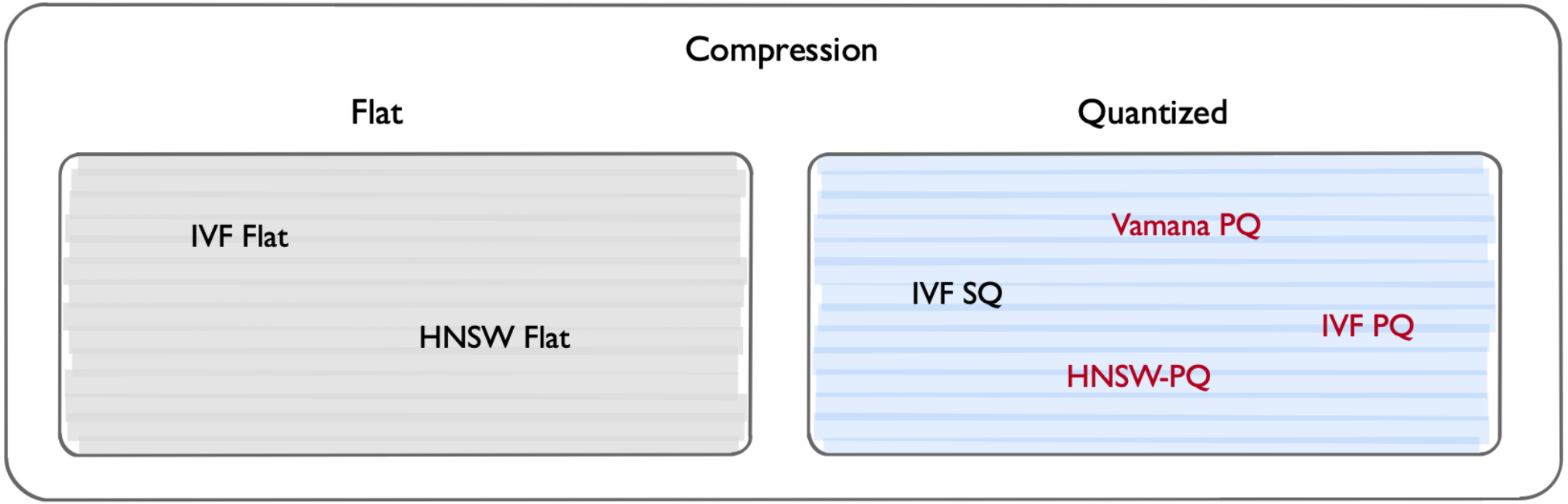
扁平化的索引 Flat indexing

- **Flat Indexing**：使用 ANN、IVF 或 HNSW 等索引，直接计算查询向量与 DB 中向量之间距离。
- 为了将其与量化变体区分开来，使用这种方式使用时通常称为 IVF-Flat、HNSW-Flat 等。



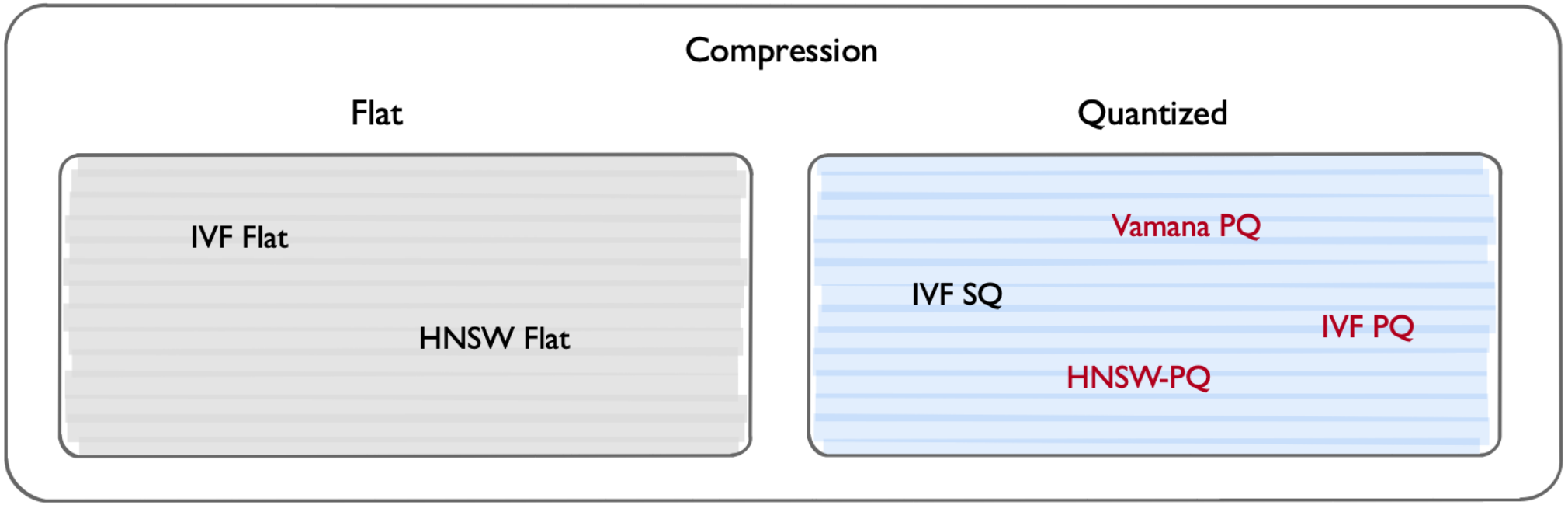
量化索引 Quantized indexing

- **量化索引**：将索引算法（IVF、HNSW）与量化方法相结合，以减少内存占用并加快索引速度。
- **量化分类**：标量量化（Scalar Quantization，SQ）或乘积量化（Product Quantization，PQ）。



量化索引 Quantized indexing

- **标量量化 SQ**：将向量对称划分为包含每个维度的最小值和最大值的容器，将向量中的浮点数转换为整数。e.g.，神经网络模型对权重参数的量化。
- **乘积量化 PQ**：考虑沿每个向量维度值分布，执行压缩和数据缩减。将较大维度向量空间分解为较小维度子空间的笛卡尔积。



3. 相似性搜索算法

原则

- 除暴力搜索能精确索引到近邻，所有搜索算法只能在**性能、召回率、内存**三者进行权衡。

向量检索算法

- 向量数据库使用近似最相邻（Approximate Nearest Neighbor，ANN），评估相似向量间相似度。
- 为了解决逐个索引向量相比，效率低问题。出现了 IVF、HNSW、LSH 等算法。

Summary

相似性搜索算法比较

分类	经典算法	召回率	索引性能
暴力搜索	K-Means	全局最优	低
基于树	KD-Trees	较高	高维下降
	Annoy	较高	较高
基于图	NSW	高	高
	HNSW	较高	较高
基于哈希	LSH	低	高
倒排索引	IVF	高	较高
量化压缩	SQ	高	低
	IQ	中	低

Reference 引用&参考

1. Maximizing the Potential of LLMs: Using Vector Databases (ruxu.dev)
2. Maglott D， Ostell J， Pruitt KD， Tatusova T. Entrez Gene: gene-centered information at NCBI. Nucleic Acids Res. 2005 Jan 1;33 (Database issue):D54-8.
3. Wang Y， Xiao J， Suzek TO， Zhang J， Wang J， Zhou Z， Han L， Karapetyan K， Dracheva S， Shoemaker BA， Bolton E， Gindulyte A， Bryant SH. PubChe m's BioAssay Database. Nucleic Acids Res. 2012 Jan;40 (Database issue):D400-12.
4. Torng W， Altman RB. 3D deep convolutional neural networks for amino acid environment similarity analysis. BMC Bioinformatics. 2017 Mar 16;18 (1):302.
5. Zheng S， Shao W， Chen L. UniVec: a database of gene expression vectors for PCA based gene similarity search. BMC Genomics. 2017 Dec 6;18 (Suppl 10):918.
6. Manning CD， Raghavan P， Schütze H. Introduction to Information Retrieval. Cambridge: Cambridge University Press， 2008.
7. Mikolov， Tomas， et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).
8. Andoni， Alexandr， and Piotr Indyk. "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions." Communications of the ACM 51.1 (2008): 117-122.
9. Jégou， Hervé， et al. "Product quantization for nearest neighbor search." IEEE transactions on pattern analysis and machine intelligence 33.1 (2010): 117-128.
10. Ge， Tiezheng， et al. "Optimized product quantization." IEEE transactions on pattern analysis and machine intelligence 36.4 (2013): 744-755.
11. Babenko， Artem， and Victor Lempitsky. "The inverted multi-index." IEEE transactions on pattern analysis and machine intelligence 37.6 (2014): 1247-1260.
12. Datar M， Immorlica N， Indyk P， Mirrokni VS. Locality-sensitive hashing scheme based on p-stable distributions. Proceedings of the twentieth annual symposium on Co mputational geometry. 2004 Jun 8:253-62.
13. Muja M， Lowe DG. Fast approximate nearest neighbors with automatic algorithm configuration. VISAPP (1). 2009 Feb 4:331-40.
14. Jégou， Hervé， et al. "Product quantization for nearest neighbor search." IEEE transactions on pattern analysis and machine intelligence 33.1 (2011): 117-128.
15. Chen， Zhenjie， and Jingqi Yan. "Fast KNN search for big data with set compression tree and best bin first." 2016 2nd International Conference on Cloud Computing an d Internet of Things (CCIOT). IEEE， 2016.
16. Dehmamy， Nima， Albert-László Barabási， and Rose Yu. "Understanding the representation power of graph neural networks in learning graph topology." Advances in Neural Information Processing Systems 32 (2019).
17. Babenko A， Lempitsky V. The inverted multi-index. IEEE transactions on pattern analysis and machine intelligence. 2014 Jun 7;37 (6):1247-60.

Reference 引用&参考

1. https://www.bilibili.com/video/BV11a4y1c7SW
2. https://www.bilibili.com/video/BV1BM4y177Dk
3. https://www.pinecone.io/learn/vector-database/
4. https://github.com/guangzhengli/ChatFiles
5. https://github.com/guangzhengli/vectorhub
6. https://www.anthropic.com/index/100k-context-windows
7. https://js.langchain.com/docs/
8. https://www.pinecone.io/learn/series/faiss/locality-sensitive-hashing/
9. https://www.pinecone.io/learn/series/faiss/product-quantization/
10. https://www.pinecone.io/learn/series/faiss/locality-sensitive-hashing-random-projection/
11. https://www.youtube.com/watch?v=QvKMwLjdK-s&t=168s&ab_channel=JamesBriggs
12. https://www.pinecone.io/learn/series/faiss/faiss-tutorial/
13. https://www.youtube.com/watch?v=sKyvsdEv6rk&ab_channel=JamesBriggs
14. https://www.pinecone.io/learn/vector-similarity/
15. https://github.com/chroma-core/chroma
16. https://github.com/milvus-io/milvus
17. https://www.pinecone.io/
18. https://github.com/qdrant/qdrant
19. https://github.com/typesense/typesense
20. https://github.com/weaviate/weaviate
21. https://redis.io/docs/interact/search-and-query/
22. https://github.com/pgvector/pgvector

Thank you

把AI系统带入每个开发者、每个家庭、每个组织，构建万物互联的智能世界

Bring AI System to every person, home and organization for a fully connected, intelligent world.

ZOMI

Course chenzomi12.github.io

GitHub github.com/chenzomi12/DeepLearningSystem