向量数据库

IVF 系列索引应用指南

腾讯云

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

IVF 系列索引应用指南

最近更新时间：2024-09-08 17:20:01

IVF 索引方式数据导入速度快，且内存空间占用低，特别**适合于亿级大规模**高维向量数据的相似性检索，本文介绍通过 Python SDK 应用 IVF 索引的操作流程。

IVF 索引介绍

IVF 索引的基本原理是将向量数据集划分为多个子集，每个子集称为一个聚类中心或一个簇。每个簇都有一个代表性的向量，称为聚类中心向量。通过构建一个倒排表，将聚类中心向量与属于该簇的向量进行关联。
在进行搜索时，首先根据查询向量找到与之最相似的聚类中心向量，然后在该聚类中心对应的倒排表中查找更接近查询向量的具体向量。这种两级索引结构可以极大地减少搜索的计算量，提高搜索效率。

使用限制

- IVF 索引需要在插入一定量的数据后才能开始训练。因此，在插入数据时无需构建索引，等待数据插入完成后需重建 IVF 索引。
- 创建集合时，需配置参数 nlist， 聚类中心的数量，建议取值范围为 `[sqrt(单分片数据量)*4, sqrt(单分片数据量)*16]` 。每个分片内至少需要写入 30*nlist 条数据，最多选取 256*nlist 条数据进行模型训练。
- 重建索引过程中，不允许数据库写入，无法停止任务。

应用示例

导入 SDK

```
import tcvectordb
from tcvectordb.model.document import Document, SearchParams, Filter
from tcvectordb.model.enum import FieldType, IndexType, MetricType,
ReadConsistency
from tcvectordb.model.index import Index, VectorIndex, FilterIndex,
IVFPQParams
```

创建客户端连接

如下示例可直接复制，运行之前，您需在文本编辑器将 api_key=A5VOgsMpGWJhUI0WmUbY********** 与 10.0.X.X 依据实际情况进行替换。

```
# 1. 创建客户端连接
client = tcvectordb.VectorDBClient(url='http://10.x.x.x',
username='root',
```

```
key='A5VOgsMpGWJhUI0WmUbY********************', timeout=30,

read_consistency=ReadConsistency.STRONG_CONSISTENCY)
```

创建 Database

```
# 2. 创建 Database
db = client.create_database(database_name='db_test_ivf')
```

创建 Collection

如下列出创建集合时，应用 IVF 系列索引需特别设置的关键参数。

参数	参数含义	取值说明
indexType	索引类型，目前支持的 IVF_FLAT、IVF_PQ、IVF_SQ4, IVF_SQ8, IVF_SQ16。	本文以 IVF_PQ 为例介绍。
nlist	在 IVF_FLAT 算法中，向量空间被划分为 nlist 个聚类中心。	建议取值范围为：[sqrt(单分片数据量)*4, sqrt(单分片数据量)*16]
M	指乘积量化中原始数据被拆分的子向量的数量。将原始数据向量拆分为 M 个子向量。每个子向量的维度为 D/M，其中 D 是原始向量的维度。然后，对每个子向量进行独立的量化，得到 M 个码本（codebook），每个码本对应一个子向量的离散化表示。最终，将 M 个码本拼接起来，得到原始向量的 PQ 编码（code）。	• 仅 IVF_PQ 类型涉及该参数，IVF 其他系列无需设置。 • M 必须能被 D（原始向量的维度）整除。

```
# 3. 创建Collection，并使用IVF_PQ索引
# 3.1  定义索引，其中向量索引为IVF_PQ
index = Index(
    FilterIndex(name='id', field_type=FieldType.String,
index_type=IndexType.PRIMARY_KEY),
    FilterIndex(name='name', field_type=FieldType.String,
index_type=IndexType.FILTER),
    VectorIndex(name='vector', dimension=128,
index_type=IndexType.IVF_PQ,
                metric_type=MetricType.COSINE, params=IVFPQParams(m=2,
nlist=10))     # nlist建议取值范围为[sqrt(单分片数据量)*4, sqrt(单分片数据
```

```
量)*16]
)

# 3.2 创建集合
coll = db.create_collection(
    name='test_ivf',
    shard=1,
    replicas=2,
    description='this is a collection of test IVF_PQ',
    index=index
)
```

写入数据

> **注意：**
> 如果创建 Collection 选择的索引类型为 IVF 系列：
> - 当第一次写入时，当前集合还没有向量索引，此时 **buildIndex** 必须为 **false**。插入原始数据之后，需通过 rebuild 训练数据并重建索引。
> - 当集合已经调用过 rebuild 创建索引后，集合已经存在向量索引，此时：
>   - 如果 **buildIndex = true**，表示新写入的数据会加入到已有的 IVF 索引中，但不会更新索引结构，此时新写入的数据可以被检索到。
>   - 如果 **buildIndex = false**，表示新写入的数据不会加入到已有的 IVF 索引中，此时新写入的数据无法被检索到。

```
# 4. 写入数据（此处写入的数据，均为随机生成的测试数据）
# 注意：IVF索引需要写入一定数据量后，才能开始训练，
#       每个分片内至少需要写入 30*nlist 条数据，最多能训练 256*nlist 条数据
data_num = 300
for i in range(data_num):
    tmp_id = str(i)
    tmp_name = ''.join(random.sample(string.ascii_lowercase, 5))
    tmp_vector = np.random.randn(1, 128)[0].tolist()
    res = coll.upsert(
        documents=[
            Document(id=tmp_id, vector=tmp_vector, name=tmp_name)
        ],
        build_index=False    # IVF索引在首次写入数据时，需要设置build_index为
False
```

```
)
```

Rebuild 索引

> **注意：**
> - 单分片内行数少于 30*nlist（nlist 为聚类中心数量）不支持训练。
> - 重建索引过程中，不允许数据库写入，无法停止任务。

如下列出 Rebuild 时需配置的关键参数，请参见下表。

参数	参数含义	配置方法及要求
drop_before_rebuild	标识在重建索引时，是否需先删除旧索引再重建新索引。 **说明：** 重建索引需要占用额外的内存空间，数据量越大，消耗的内存空间越大。在重建索引之前，您需根据实际资源情况选择是否需先删除旧索引再重建，避免引起内存占满而阻塞业务正常运行。	取值如下所示： • **True**：重建之前，先删除旧索引再重建索引。 **说明：** 内存资源不足时，可先删除旧索引，在新索引还没有创建完成之前，该表无法正常读写。 • **False**：重建之前，不删除旧索引，创建新索引完成之后再删除旧索引。默认为 False。 **说明：** 内存资源足够的情况下，可不删除旧索引。在新索引还没有创建完成之前，该表可读，禁止写入。
throttle	标识是否限制构建索引的单节点 CPU 核数。 **说明：** 重建索引会消耗 CPU 资源，为防止资源打满影响写入或者检索等操作，请根据业务实际配置重建索引的 CPU 核数。	取值如下所示： • **0**：不限制 CPU 核数。在模型训练期间，会消耗大量的 CPU 资源。重建索引任务将会尽快执行，但可能会对其他集合的读写操作产生影响。 • **1**：CPU 核数为 1，即仅使用 CPU 1 核进行模型训练，可避免构建索引期间对其他集合产生影响，但任务执行较慢。

```
# 5. 重建索引
# dropBeforeRebuild：标识重建索引之前，是否需要先删除旧索引，再创建新索引。
# throttle：标识是否限制构建索引的单节点 CPU 核数。默认为 1，即使用1核进行训练；可设置为 0，表示不限制 CPU 核数。
coll.rebuild_index(drop_before_rebuild=False, throttle=1, timeout=30)
```

查看索引状态

```
# 6. 查询索引是否重建完成
db = client.database('db_test_ivf')
res = db.describe_collection('test_ivf')
print(vars(res))
```

返回参数 **indexStatus** 中的 **status** 标识当前 Collection 重建索引的状态，**startTime** 显示重建索引开始的时间。

- **ready**：表示当前 Collection 准备就绪，可正常使用。
- **training data**：IVF 索引下特定状态，表示当前 Collection 正在进行数据训练，即训练模型已生成向量数据。
- **building index**：表示当前 Collection 正在重建索引，即将生成的向量数据存储到新的索引中。
- **failed**: 重建索引失败，可能会影响集合读写操作。

> ⚠ **注意：**
> **training data** 与 **building index** 状态期间不可写入数据。若重建索引之前先删除旧索引，则集合不可进行相似性查询，只能进行精确查询。

```
{
    "code": 0,
    "msg": "operation success",
    "collection": {
        "database": "db_test_ivf",
        "collection": "test_ivf",
        "documentCount": 4,
        "indexes": [
            ......
        ],
        "indexStatus": {
            "status": "ready",
```

```
            "startTime": ""
        }
    }
}
```