Python使用Faiss库实现向量近邻搜索

原创 蚂蚁 蚂蚁学Python 8月2日

本文是Python应用于推荐系统领域的技术文章。

Embedding的近邻搜索是当前图推荐系统非常重要的一种召回方式，通过item2vec、矩阵分解、双塔DNN等方式都能够产出训练好的user embedding、item embedding，对于embedding的使用非常的灵活：

- 输入user embedding，近邻搜索item embedding，可以给user推荐感兴趣的items
- 输入user embedding，近邻搜搜user embedding，可以给user推荐感兴趣的user
- 输入item embedding，近邻搜索item embedding，可以给item推荐相关的items

然而有一个工程问题，一旦user embedding、item embedding数据量达到一定的程度，对他们的近邻搜索将会变得非常慢，如果离线阶段提前搜索好在高速缓存比如redis存储好结果当然没问题，但是这种方式很不实时，如果能在线阶段上线几十MS的搜索当然效果最好。

Faiss是Facebook AI团队开源的针对聚类和相似性搜索库，为稠密向量提供高效相似度搜索和聚类，支持十亿级别向量的搜索，是目前最为成熟的近似近邻搜索库。

接下来通过jupyter notebook的代码，给大家演示下使用faiss的简单流程，内容包括：

1. 读取训练好的Embedding数据
2. 构建faiss索引，将待搜索的Embedding添加进去
3. 取得目标Embedding，实现搜索得到ID列表
4. 根据ID获取电影标题，返回结果

1、读取预训练好的Embedding数据

```
import pandas as pd
import numpy as np

```

```
df = pd.read_csv("./datas/movielens_sparkals_item_embedding.csv")

# 提取要使用的电影ID列表，注意要转换成int64
ids = df["id"].values.astype(np.int64)

# 记录ID列表的大小
ids_size = ids.shape[0]

# 读取内容embedding数据，转换成二维array
import json
import numpy as np
datas = []
for x in df["features"]:
    datas.append(json.loads(x))
# 变成二维array
datas = np.array(datas).astype(np.float32)

# 记录数据维度
dimension = datas.shape[1]
```

2、使用faiss建立索引

```
import faiss

index = faiss.IndexFlatL2(dimension)
index2 = faiss.IndexIDMap(index)

index2.add_with_ids(datas, ids)
```

3、实现近邻搜索

```
# 读取user embedding数据
df_user = pd.read_csv("./datas/user_embedding.csv")
df_user.head()

# 挑选一条user的embedding，转换成1行N列的二维array
user_embedding = np.array(json.loads(df_user[df_user["id"] == 10]["features"]
user_embedding = np.expand_dims(user_embedding, axis=0).astype(np.float32)
```

```
user_embedding

# 实现搜索，这里的I就是近邻ID列表
tok = 30
D, I = index.search(user_embedding, topk)     # actual search
```

4、拼接得到内容列表

```
# 把搜索出来的ID变成Series
target_ids = pd.Series(I[0], name="MovieID")

# 读取电影数据
df_movie = pd.read_csv("./datas/ml-1m/movies.dat",
                       sep="::", header=None, engine="python",
                       names = "MovieID::Title::Genres".split("::"))
# 实现内容JOIN
df_result = pd.merge(target_ids, df_movie)
```

df_result就是最终可以返回给前端的推荐的电影ID列表了。

如果本文对你有帮助的话，帮忙点个赞吧^_^

代码：https://github.com/peiss/ant-learn-recsys