实验 2 豆瓣电影数据的知识感知推荐

实验背景

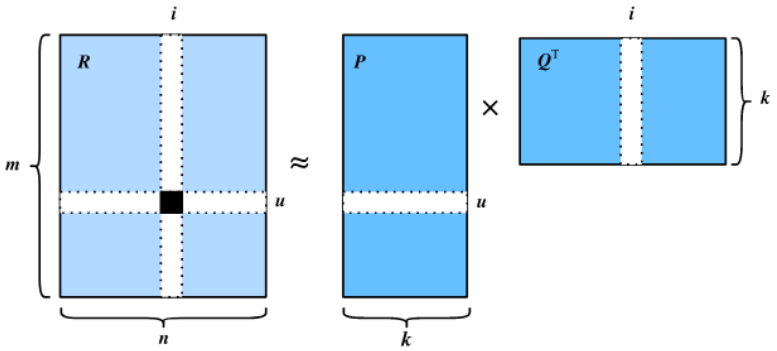
知识图谱（KG）在提高推荐的准确性和可解释性方面显示出了巨大的潜力。KG 中丰富的实体和关系信息可以强化用户和物品之间的关系建模，因为它们不仅揭示了物品之间的各种相关性（如两部电影由同一个人导演），还可以用来解释用户偏好（如将用户对电影的选择归因于其导演）。

在本次实验中，我们要求各位同学从公开图谱中匹配指定电影对应的实体，并抽取合适的部分图谱，按照规则对抽取到的图谱进行处理（Stage1）；进而，基于对实验一中的豆瓣电影评分数据，结合 Stage1 所获得的图谱信息，进行可解释的、知识感知的个性化电影推荐（Stage2）。

实验介绍

在本次实验中，我们会提供基于实验一中电影评分数据生成的训练集和测试集，以及 baseline (MF)的代码，要求将 Stage1 所获得的图谱整合到训练数据中，并基于 baseline，完成基于图谱嵌入的和基于 GNN 的知识感知推荐。分析不同的设计（不同的图谱嵌入方法、不同的训练方式、不同的图卷积聚合方式以及图卷积层的数量等）对知识感知推荐性能的影响，同时需要对比分析知识感知推荐与 MF 的实验结果。

矩阵分解 MF 是推荐系统中的基础算法，其在 2006 年举行的 Netflix 竞赛发挥了关键作用。该模型将用户-物品的交互矩阵 R 分解为用户的潜在矩阵 P 和物品的潜在矩阵 Q。



其中 Q 的第 i 行q_i代表物品 i 的潜在特征，P 的第 u 行p_u代表用户 u 对物品相应潜在特征的感兴趣程度。因此可以通过二者的内积

$$\hat{y}_{ui} = \mathrm{p}_u \mathrm{q}_i^{\mathsf{T}}$$

来预测用户 u 对物品 i 的偏好程度。

在代码层面，一般通过 nn.Embedding(n_users/n_items, embed_dim)来创建用户/物品的潜在矩阵，其中 n_users/n_items 为用户/物品的数量, embed_dim 为潜在

特征的维度。然后选择 BPR Loss（贝叶斯个性化排序损失）来优化 MF 模型，它认为用户喜爱的物品 i 应该比不喜爱的（或未交互过的）物品 j 有更高的预测得分，可以看出 BPR Loss 的训练数据由正负样本对（i，j）组成，其数学表达为，

$$\mathcal{L}_{\text{BPR}} = -\sum_{(u,i,j\in D)} \ln \sigma\left(\hat{y}_{ui} - \hat{y}_{uj}\right)$$

其中 $D = \{(u,i,j) \mid i \in I_u^+,\ j \in I \backslash I_u^+\}$ 是训练集，I_u^+表示用户 u 喜爱的物品集合，而 $I \backslash I_u^+$ 表示除用户 u 喜欢物品之外的所有其他物品的集合；$\hat{y}_{ui}$和$\hat{y}_{uj}$分别用户 u 对物品 i 和物品 j 的预测得分；$\sigma()$指 sigmoid 函数。

有关矩阵分解的理论部分可参考第 8 节个性化检索部分的相关内容，矩阵分解和 BPR 损失的代码教程可参考相关介绍文章[1]。

在第一阶段中，我们已经从 Freebase 中抽取出包含 578 部电影的小规模图谱。在此次的第二阶段，我们提供了由实验一中的评分数据得到的训练集、测试集和矩阵分解的代码（包括数据加载，模型搭建和模型训练三个部分），本次实验将围绕这些信息进行展开，详述如下：

第二阶段任务：图谱推荐

在我们给出的训练集文件 train.txt 和测试集文件 test.txt 中，提供了每个用户打分≥4 的电影集合，这些电影被视为该用户的正样本，其中每一行对应一个用户，每一行的第一个值为该用户的 ID，余下的值为该用户的正样本 ID 集合。此外我们将用户的 ID 和电影的 ID 映射到从 0 开始的索引值，映射关系分别保存在 user_id_map.txt 和 movie_id_map.txt 这两个文件中。通过图谱实体 ID 到电影 ID 之间的映射关系（douban2fb.txt）以及电影 ID 到从 0 开始的索引值之间的映射关系（movie_id_map.txt），第一阶段抽取的电影图谱能够轻松地整合到推荐系统中。

第二阶段（Stage 2）的实验内容包含以下部分：

[1] 【必做】根据映射关系，将电影实体的 ID 映射到$[0, num\ of\ movies)$范围内。将图谱中的其余实体映射到$[num\ of\ movies,\ num\ of\ entities)$范围内，将关系映射到 $[0, num\ of\ relations)$范围内。再根据这些映射关系，将第一阶段获得的电影图谱映射为由索引值组成的三元组，即（头实体索引值，关系索引值，尾实体索引值），并保存到 stage2\data\Douban\kg_final.txt 文件中。

[2] 【必做】熟悉 baseline 的框架代码，包括数据加载部分（stage2\data_loader 文件夹下的 loader_base.py 和 loader_KG_free.py），模型搭建部分（stage2\model 文件夹下的 KG_free.py），以及模型训练部分（stage2 文件夹下的

[1] https://d2l.ai/chapter_recommender-systems/index.html

main_KG_free.py）

说明：

- 我们提供的是基础 MF 算法的代码，但大家可以根据自己掌握的情况选择合适的 MF 算法，基本的 MF、NMF 和 PMF 都是可以的，额外的约束也自选。如果实验一用的就是矩阵分解方法，也可以直接调用实验一的对应数据。

[3] 【必做】基于 baseline 框架代码，完成基于图谱嵌入的模型，包括数据加载部分（stage2\data_loader 文件夹下的 loader_Embedding_based.py）和模型搭建部分（stage2\model 文件夹下的 Embedding_based.py）的相关代码模块：

a) 在 loader_Embedding_based.py 中按要求实现 KG 的构建。

b) 在 Embedding_based.py 中实现 TransE，TransR 算法，并尝试通过相加，逐元素乘积，拼接等方式为物品嵌入注入图谱实体的语义信息。

c) 采用多任务方式（KG 损失与 CF 损失相加）对模型进行优化。

d) 将多任务方式更改为迭代优化方式，即 KG 损失与 CF 损失迭代地对模型进行优化。

e) 【选做】调研相关综述[2]，思考如何改进自己的模型，再动手尝试一下。

[4] 【必做】基于 baseline 框架代码，完成基于 GNN 的模型，包括数据加载部分（stage2\data_loader 文件夹下的 loader_GNN_based.py）和模型搭建部分（stage2\model 文件夹下的 GNN_based.py）的相关代码模块：

a) 在 loader_Embedding_based.py 中按要求实现 KG 的构建和归一化拉普拉斯矩阵的计算。

b) 在 GNN_based.py 中实现 TransE，TransR 算法；完成图卷积模块，中心节点表征与一跳领域表征三种聚合方式的代码。

c) 源代码采用 KG 损失与 CF 损失迭代更新的方式，要求将其改为多任务方式，即将 KG 损失与 CF 损失相加，使用总体的损失进行模型优化。

d) 【选做】调研上述综述，思考如何改进自己的模型，再动手尝试一下。

说明：

- 图卷积和聚合操作的相关说明可参考 KGAT[3]

[5] 【必做】本次实验的评价指标采用 Recall@5，NDCG@5，Recall@10 和 NDCG@10。需要分析不同的设计（不同的图谱嵌入方法、不同的训练方式、不同的图卷积聚合方式以及图卷积层的数量等）对知识感知推荐性能的影响，同时需要对比分析知识感知推荐与 MF 的实验结果。

---

[2] https://ieeexplore.ieee.org/abstract/document/9216015

[3] https://dl.acm.org/doi/abs/10.1145/3292500.3330989

实验环境说明

本次实验建议在 anaconda 的虚拟环境下进行，依赖的 python 包有 pytorch（cpu 版本也可以），tqdm，numpy，pandas，scikit-learn。同学们在安装完 anaconda 后，可以通过以下几行命令完成本次实验的环境配置，

1. 创建并激活新环境

   ```
   conda create -n web_exp python=3.7
   conda activate web_exp
   ```

2. 安装 pytorch cpu 版本（有条件的同学也可以安装 gpu 版本的）

   ```
   conda install pytorch==1.8.0 torchvision==0.9.0 torchaudio==0.8.0 cpuonly -c pytorch
   ```

3. 安装其它的依赖包

   ```
   conda install tqdm numpy pandas scikit-learn
   ```

数据集说明

我们提供了以下文件，包括：

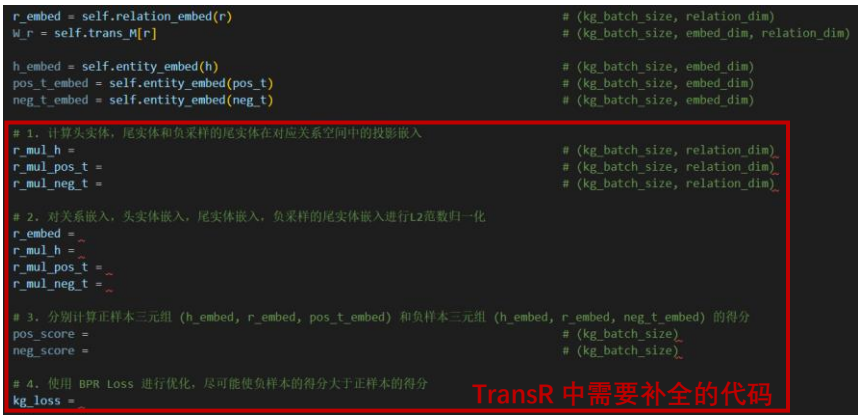
训练集 train.txt 和测试集 test.txt,每一行对应一个用户，其中第一个值为该用户的索引值，后面若干个值为该用户打分≥4 的电影索引值集合，被视为该用户的正样本集合，注意索引值都是从 0 开始编号的，统计信息如下：

训练集train.txt 中:	测试集test.txt 中:
用户数量: 447, 电影数量: 578, 训练集大小: 41830	用户数量: 447, 电影数量: 574, 训练集大小: 10840
交互矩阵的稀疏度: 83.81%	交互矩阵的稀疏度: 95.78%

注意：训练集和测试集均放在 stage2\data\Douban\文件夹下。

电影 ID 到索引值之间的映射关系 movie_id_map.txt，其中第一列为豆瓣电影 ID，第二列为其对应的索引值。结合图谱实体 ID 到电影 ID 之间的映射关系 douban2fb.txt，可以将电影实体 ID 映射到$[0, num\ of\ movies)$范围内。

Baseline 文件夹 stage2，包含 baseline 模型完整的框架流程，需要同学们基于 baseline，完成基于嵌入的和基于 GNN 的知识感知推荐，需要补全的模块在代码中均有注释提示，按要求补全代码即可。TransR 的示例片段如下图所示：



所涉及的各种数据和代码，可在如下地址进行下载（密码：web2022）：https://rec.ustc.edu.cn/share/b0c73a20-736f-11ed-8663-a59f82219bf7

实验要求

本次实验要求分组完成，每组最多 3 人（可以少于 3 人，但无优惠政策）。

实验持续时间约为 4-5 教学周，实验报告的具体提交时间和更多详细要求将于第二阶段公布。

提交说明

请于截止日期（2023 年 1 月 1 日晚 23：59）前将实验二完整的实验报告（整个实验提交一份报告即可）提交到课程邮箱 ustcweb2022@163.com，具体要求如下：

1. 邮件标题以及压缩包命名为"组长学号-组长姓名-实验 2"格式。邮件正文中请列出小组所有成员的姓名、学号。
2. 因未署名造成统计遗漏责任自行承担，你可以将邮件抄送你的队友。
3. 实验报告请务必独立完成，如果发现抄袭按 0 分处理。
4. 迟交实验将不被接收。
5. 后续版本会进一步更新具体实验报告要求。