Web信息处理与应用

实验二 第2阶段

知识感知推荐——图谱推荐

助教组　　2023.12.4

- **豆瓣电影数据的知识感知推荐**

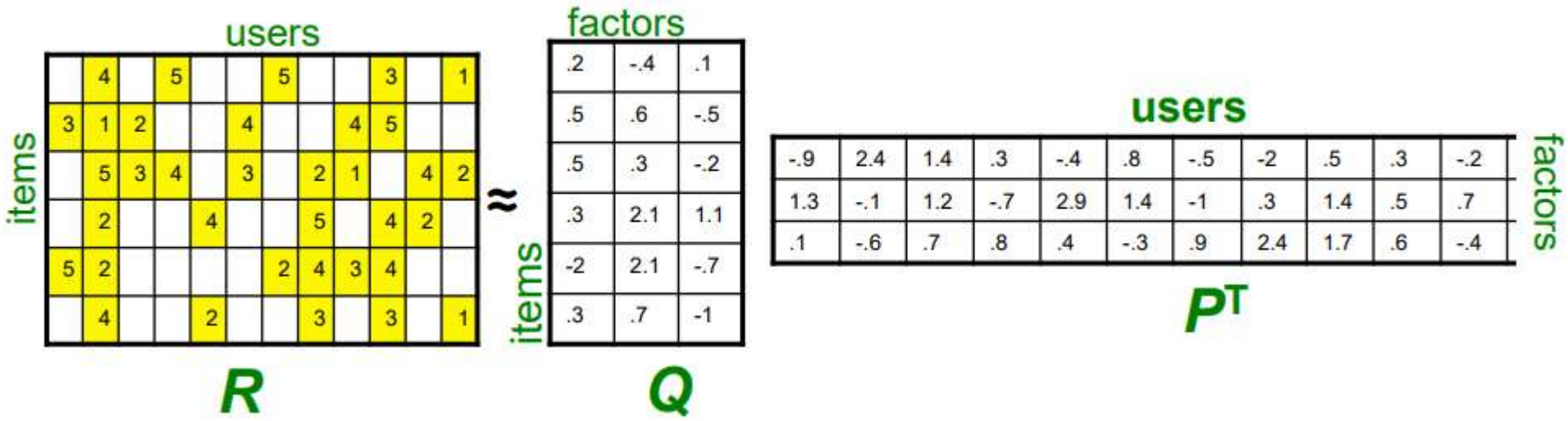
  - 在Stage1中，我们已经从公开图谱中匹配指定电影对应的实体，并抽取合适的部分图谱，按照规则对抽取到的图谱进行了处理（Stage1）
  - 在Stage2，我们要基于对实验一中的豆瓣电影评分数据，结合Stage1所获得的图谱信息，进行可解释的、知识感知的个性化电影推荐（Stage2）

- **第二阶段任务：图谱推荐**

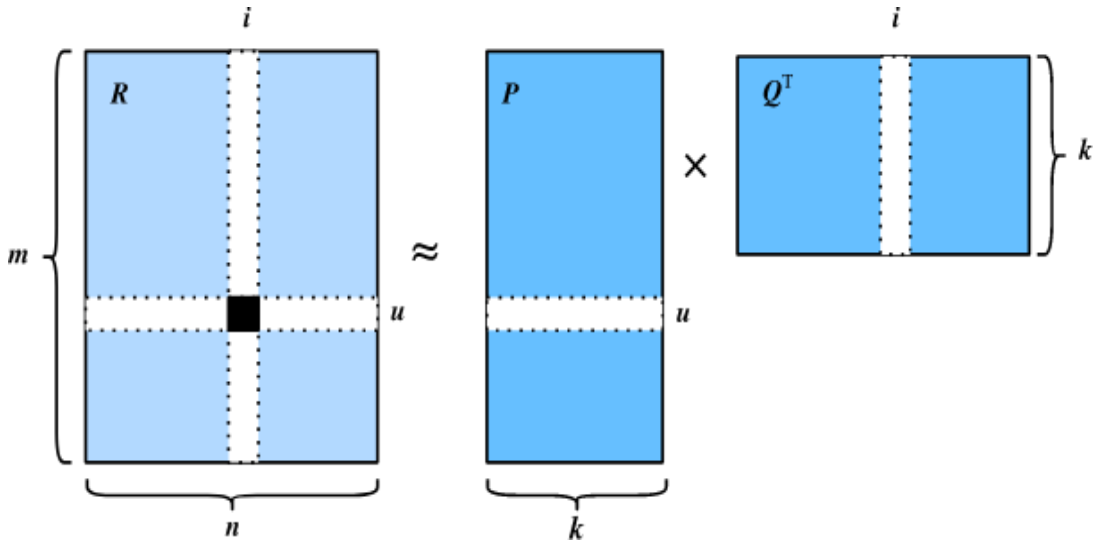
  - 在第一阶段中，我们已经从 Freebase 中抽取出包含 578 部电影的小规模图谱。
  - 在本次实验中，我们会提供基于实验一中电影评分数据生成的训练集和测试集，以及 baseline (MF)的代码，要求将 Stage1 所获得的图谱整合到训练数据中，并基于 baseline，完成基于图谱嵌入的知识感知推荐。分析不同的设计（不同的图谱嵌入方法、不同的训练方式等）对知识感知推荐性能的影响，同时需要对比分析知识感知推荐与 MF 的实验结果。

- **第二阶段任务：图谱推荐**

  - 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。



- **第二阶段任务：图谱推荐**



- 其中 Q 的第 i 行qi代表物品 i 的潜在特征，P 的第 u 行pu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 $\hat{y}_{ui} = \mathrm{p}_u \mathrm{q}_i^T$ 来预测用户 u 对物品 i 的偏好程度。

- **第二阶段任务：图谱推荐**

  - 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

- **第二阶段任务：图谱推荐**

  - BPR损失的数学表达式为，

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

  - 其中$D = \{(u,i,j) \mid i \in I_u^+, j \in I \setminus I_u^+\}$是训练集，$I_u^+$表示用户u喜爱的物品集合，而$I \setminus I_u^+$表示出用户u喜欢物品之外的所有其他物品的集合；$\hat{y}_{ui}$和$\hat{y}_{uj}$分别表示用户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https://d2l.ai/chapter_recommender-systems/index.html 。

- **第二阶段任务：图谱推荐**

  - 第二阶段（Stage2）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。

- **第二阶段任务：图谱推荐**

  - 第二阶段（Stage2）的实验内容包含以下部分：
    - [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main_KG_free.py）

- **第二阶段任务：图谱推荐**

- 第二阶段（Stage2）的实验内容包含以下部分：
  - [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chapter12中介绍的 TransE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(有兴趣的同学可以采用TransR等算法)。
    - c)　按照给出的源代码，采用多任务方式（KG 损失与 CF 损失相加）对模型进行更新。（原始框架已经按照多任务方式组织好，补全代码后训练即可。）

- **第二阶段任务：图谱推荐**

  - 第二阶段（Stage2）的实验内容包含以下部分：
    - [4] **【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的图谱嵌入方法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。

- **第二阶段任务：图谱推荐**

- 第二阶段（Stage2）的实验内容包含以下部分：
  - [5]【选做】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) 按照给出的源代码，采用 KG 损失与 CF 损失迭代更新的方式对模型进行更新。（原始框架已经按照迭代更新方式组织好，补全代码后训练即可。）

• 第二阶段任务：图谱推荐

- 第二阶段（Stage2）的实验内容包含以下部分：
  - [6]【选做】探究不同的训练方式对知识感知推荐性能的影响。（比如多任务方式和迭代优化方式）
  - [7]【选做】调研相关综述，思考如何改进自己的模型，再动手尝试一下。
- 【选做】仅作为兴趣探索，不影响最终分数，时间不充裕情况下可以不做

- **第二阶段任务：图谱推荐**

  - 说明及技巧：
    - 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，额外的约束可自选，也可以选择调用实验一使用的方法。
    - 图卷积和聚合操作的相关说明可参考 KGAT，https://dl.acm.org/doi/abs/10.1145/3292500.3330989
    - 相关综述可参考：https://ieeexplore.ieee.org/abstract/document/9216015

- **安装环境**

- 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

- **我们提供了必要的数据文件，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：

用户数量：447，电影数量：578，训练集大小：41830

交互矩阵的稀疏度：83.81%

测试集test.txt 中：

用户数量：447，电影数量：574，训练集大小：10840

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。

数据集说明

- **我们提供了必要的数据文件，包括：**

  - **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。

4164444	0
4268598	1
4023638	2
1305903	3
3546019	4
3313801	5
3742360	6
1978709	7
3148748	8
1765009	9

数据集说明

- **我们提供了必要的数据文件，包括：**

  - **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（示例片段见下一页）。

- **我们提供了必要的数据文件，包括：**

  - **Baseline 文件夹 stage2**
  - TransE 的示例片段如下图所示，其中红框为需要大家补全的代码：

```
def calc_kg_loss_TransE(self, h, r, pos_t, neg_t):
    """
    h:      (kg_batch_size)
    r:      (kg_batch_size)
    pos_t:  (kg_batch_size)
    neg_t:  (kg_batch_size)
    """
    r_embed = self.relation_embed(r)                                          # (kg_batch_size, relation_dim)

    h_embed = self.entity_embed(h)                                            # (kg_batch_size, embed_dim)
    pos_t_embed = self.entity_embed(pos_t)                                    # (kg_batch_size, embed_dim)
    neg_t_embed = self.entity_embed(neg_t)                                    # (kg_batch_size, embed_dim)

    # 5. 对关系嵌入，头实体嵌入，尾实体嵌入，负采样的尾实体嵌入进行L2范数归一化
    r_embed =
    h_embed =
    pos_t_embed =
    neg_t_embed =

    # 6. 分别计算正样本三元组 (h_embed, r_embed, pos_t_embed) 和负样本三元组 (h_embed, r_embed, neg_t_embed) 的得分
    pos_score =                                                               # (kg_batch_size)
    neg_score =                                                               # (kg_batch_size)

    # 7. 使用 BPR Loss 进行优化，尽可能使负样本的得分大于正样本的得分
    kg_loss =

    l2_loss = _L2_loss_mean(h_embed) + _L2_loss_mean(r_embed) + _L2_loss_mean(pos_t_embed) + _L2_loss_mean(neg_t_embed)
    loss = kg_loss + self.kg_l2loss_lambda * l2_loss
    return loss
```

数据集说明

- **我们提供了必要的数据文件，包括：**

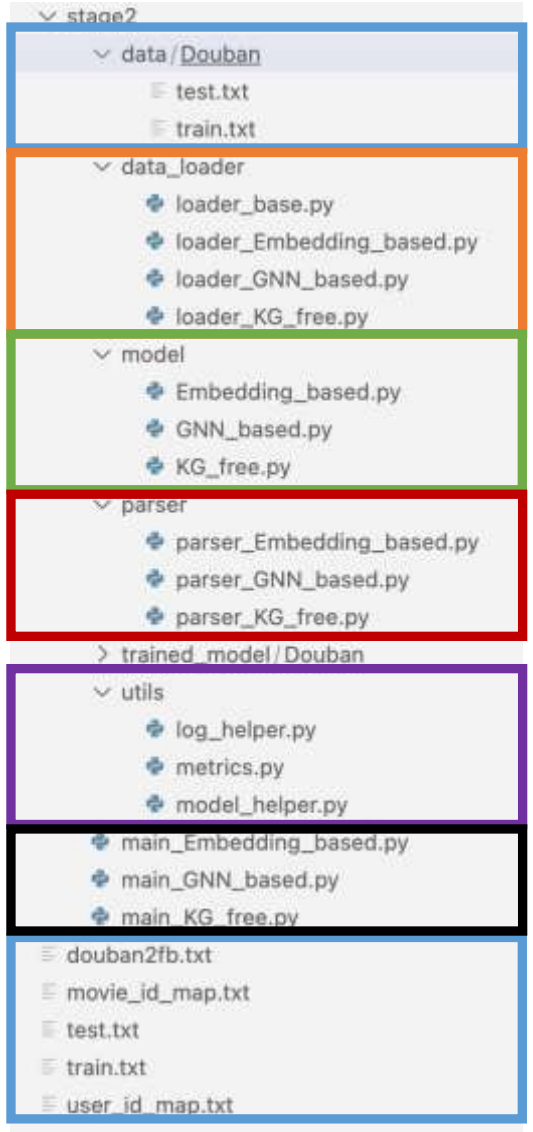
  - 以上所涉及的所有数据和代码，均可在以下地址进行下载：
  - 链接： https://rec.ustc.edu.cn/share/154c4c60-919e-11ee-a7c6-ed52c6bd9313
  - 密码： web2023

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）
- 请于截止日期（2023年12月24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，具体要求参见实验二说明文档

- **我们提供了本次实验的框架，需要在指定位置补全代码**

数据



数据

数据加载

模型

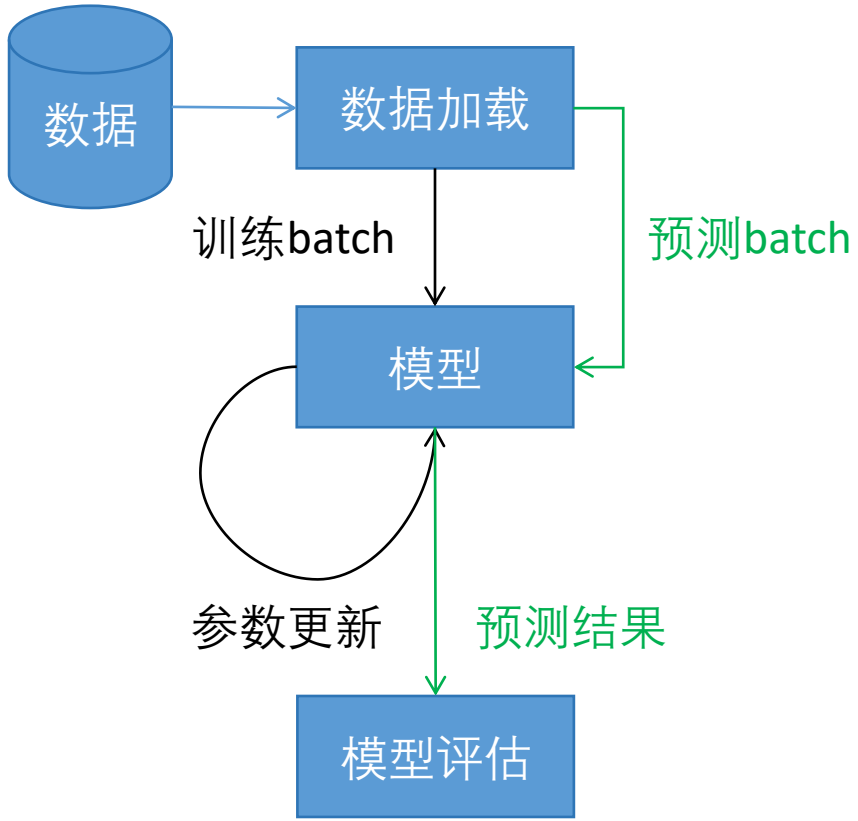
需要补全的代码

（命令行）参数设置

辅助函数（日志、指标计算等）

执行入口文件

框架流程



- **数据**

user_id_map.txt
用户UID 到 索引值的映射

```
1    1386692 0
2    1450280 1
3    4012280 2
4    1128980 3
5    1427432 4
6    2683561 5
7    1112625 6
8    1138893 7
9    1552171 8
10   1709040 9
11   1794736 10
12   1243334 11
13   1476533 12
14   1382659 13
15   1932734 14
```

movie_id_map.txt
电影MID 到 索引值的映射

```
1    4164444 0
2    4268598 1
3    4023638 2
4    1305903 3
5    3546019 4
6    3313801 5
7    3742360 6
8    1978709 7
9    3148748 8
10   1765009 9
11   1291575 10
12   1792928 11
13   3066739 12
14   1291574 13
15   3287562 14
```

train.txt / test.txt
用户-电影交互记录

```
1    0 1 2 5 6 10 14 17 19 23 26 32 33 38 42 45 49 50 51 55 57
2    1 3 8 17 22 23 26 35 42 43 44 45 50 52 55 57 59 66 67 71 8
3    2 1 7 8 15 18 21 22 23 26 28 30 31 32 47 53 61 62 66 77 84
4    3 256 258 6 270 277 280 537 282 541 542 33 548 549 553 298
5    4 512 514 3 523 524 527 531 532 536 25 540 541 29 30 546 3
6    5 32 7 392 492 366 433 465 407 569 443 349
7    6 512 515 518 519 8 9 522 11 520 523 16 18 532 21 534 25 5
8    7 3 8 524 17 18 532 21 529 535 536 542 30 543 546 35 548 3
9    8 3 264 525 529 21 280 282 284 285 287 546 291 292 293 550
10   9 519 12 524 14 526 18 21 22 535 23 26 28 30 32 33 548 550
11   10 0 1 6 264 527 275 23 28 541 30 286 33 548 38 39 295 565
12   11 514 515 6 7 522 10 527 530 21 542 543 33 546 35 548 549
13   12 256 2 5 6 7 262 264 528 18 270 277 278 283 540 284 30 3
14   13 4 6 7 10 14 15 17 18 530 283 30 543 33 290 38 46 306 52
15   14 13 528 533 538 540 38 560 561 60 65 66 74 78 85 88 105
```

用户索引值　　电影索引值