

深度学习期末报告

Ljl zhy

【摘要】这次大作业我们小组主要尝试了两种模型，GCE-GNN 和 FMLP-Rec 对短序列推荐问题在 Beauty 和 Cell 数据集上进行了对比实验。通过预实验的数据筛选模型并确定模型，并且根据先验知识为模型添加了类似于专家系统的模块，然后对模型参数的调整优化模型，最终得到测试集上的预测结果。

一、问题描述和定义

序列推荐 (Sequential Recommendation, SR) 是推荐系统中一项非常重要的任务，它通过对用户行为的序列，比如购买商品的序列进行建模，从而学到用户兴趣的变化，进而能够对用户下一个行为进行预测。

短序列推荐 (Session-based Recommendation, SBR) 相当于是序列推荐中的一个子任务，它和序列推荐最大的不同是，短序列推荐中用户交互的数据往往偏少。解决的是匿名用户的推荐问题，也就是在网络当中建立一次会话当中生成的即时推荐，换句话说就是不知道用户是谁，这种方法可以对一些不需要登陆，或者无法记录用户浏览行为的网站快速产生有效的推荐。

经过对训练集和测试集数据的分析，发现训练集和测试集中的交互序列长度都比较小，因此这次大作业需要解决的问题就是一个短序列推荐的问题。

二、方法介绍

Session-based Recommendation 推荐本质上是一个 Next-Item-Prediction 任务，因为用户交互的序列非常短，根据训练集的数据，序列长度一般在 2-6 之间，没有办法通过长序列交互来学习到用户的长期兴趣。

在建模形式上，除了传统通过马尔可夫转移矩阵来建模的 FPMC 模型，其余模型都是基于深度学习上建立的。在深度学习的模型中又分为两大主要的阵营。

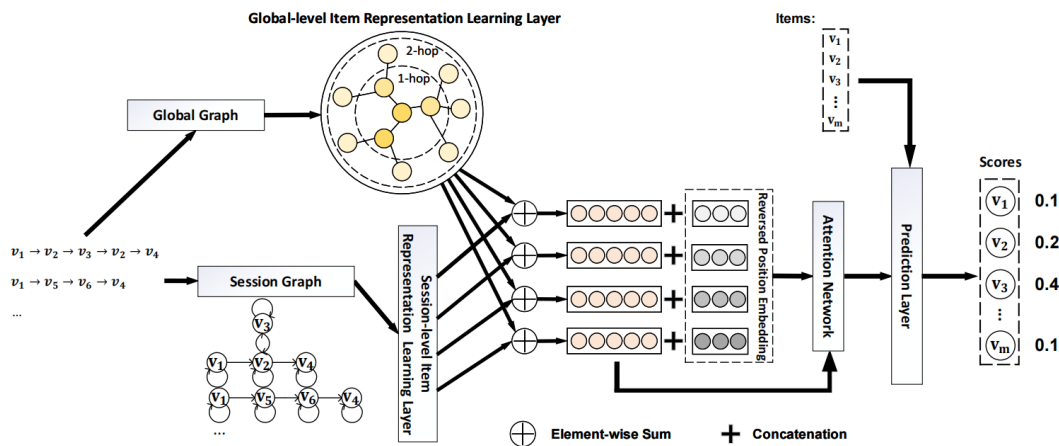
一方是基于 RNN 循环神经网络，对每一个连续的序列进行建模，包括 GRU4REC 和 NARM，其中 NARM 注意力机制运用到了对 session 的建模中，来实现局部信息和全局信息的综合考虑。

另一方是基于 GNN 图神经网络，包括 SR-GNN，GCE-GNN，GC-SAN 和 CA-TAN，总体上大同小异，基本思路都是通过图神经网络来获得物品表示，然后计算会话的局部表示（当前兴趣）和全局表示（长期兴趣）并结合起来获得最终的会话表示。

在我们的项目中，主要对比了深度学习模型中基于多层感知机和基于 GNN 两种方向的模型，最终选用基于 GCE-GNN 的方法。

（一）GCE-GNN 模型

GCE-GNN 模型采用的是图神经网络来对序列进行建模。GCE-GNN 第一次引入 global-level 物品转移图(pairwise item-transition graph)来进行序列推荐，该图是从所有 session 的物品转移数据中构造而来的。并且从两个维度考虑 pairwise item-transition 关系，包括 global-level item-transition graph 和 session-level item-transition graph。针对不同的图，提出了不同的表征方法。其中，global-level 提出了 session-aware 注意力机制，能够选择性地从全局图中抽取和与目标 session 序列相关的信息。session-level 则提出了一种 position-aware 的注意力机制来挖掘逆序位置信息，并融入到 item 的表征中，这个机制对序列推荐性能的提升非常重要。



模型结构如上图所示：

1. 局部图和全局图的创建层：局部图是一个有向图。包括了 4 种边关系，入度边，出度边，同时入度和出度边(两个结点互相指向)，自连接边。如图中左下角的部分，每条序列构造连边的时候，根据相邻物品结点构造转移边以及自连接边。其中，相邻结点的转移边又可以根据两个结点之间的关系区分为，仅入度边，仅出度边，同时入度和出度边（两个结点互相指向）。全局图是一个带权无向图。作者定义了一个较小的距离，任意两个结点想构造连边时，这两个结点之间的单位距离必须小于这个较小的距离。构造的全局图是一个带权的无向图，连边权重使用共现的次数来表示。对于每个结点，只保留 Top-N 权重的边。

2. Embedding 层：经过建模以后的图通过 Embedding 层，该层分别提取了一个结点在 session-graph 图中和在 global-graph 图中的表征，并将其融合在一起，得到单个物品的表征。

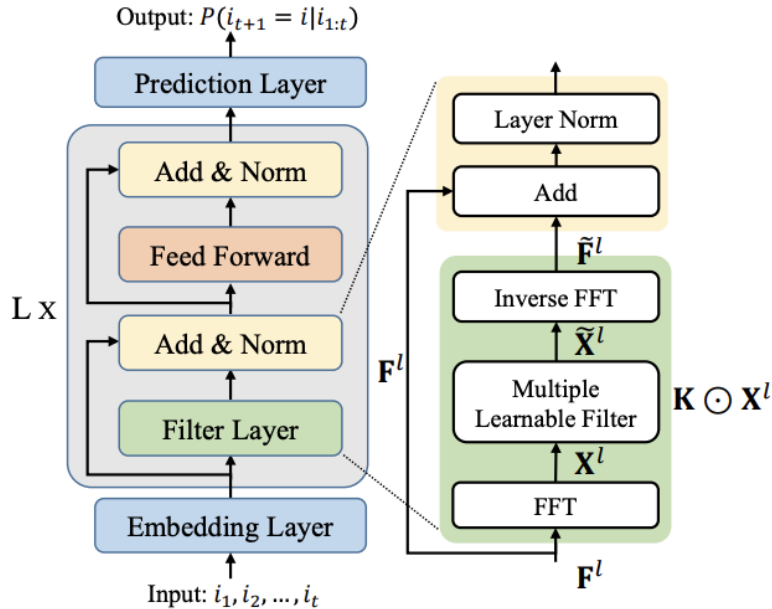
3. 注意力机制层：为了能够衡量全局图上的信息是否和目标 session 序列相关，以及是否对目标结点的表征有作用，作者提出了一种 session-aware 的注意力机制，计算了 global-graph 上和 v_i 相邻的结点 v_j 的贡献值 $\pi(v_i, v_j)$ ：

$$\pi(v_i, v_j) = \text{softmax}(q_i^T \text{LeakyRelu}(W_1[s \cdot h_{v_j} \parallel w_{ij}]))$$

4. 预测层：通过 Prediction Layer 得到的各个 item 的概率进行一次 Softmax 得到最终的每个 item 的 score 并且将得分进行排序得到最终的预测结果。

（二）FMLP-Rec 模型

FMLP 摒弃了复杂的 RNN 和 GNN 结构，采用了深度学习中最简单的纯 MLP 架构，通过添加滤波器来避免因为过多的参数量导致的模型过拟合的问题。滤波器层是通过在频域下的矩阵点乘来实现的，其等价于时域中的循环卷积，能够在时间复杂度更低的情况下提供更大的感受野。通过一个纯 MLP 架构的模型以进行用户历史序列的编码并完成推荐任务。由于该模型是在频域下利用 MLP 进行序列编码，所以模型的时间复杂度远远低于 Transformer 结构，但是却有着相同的感受野。



模型结构如上图所示，模型主要由三部分组成：

1. 编码层：Embedding 层，这一步是将所有的 item 和 position 都进行了编码，最终将

item_embedding 和 position_embedding 结合在一起得到最终的 sequence_embedding, 包含了物品的信息和物品所在序列的顺序信息。

2. 可学习的滤波模块层：这一层是由若干个可学习的滤波模块构成的，每一个模块包括两个小部分：Filter Layer（滤波层）和 Intermediate（前馈层）。

3. 预测层：通过将学习后的滤波模块层得到的 Sequence_embedding 和物品的 item_embedding 进行相似度计算，得到预测的排序。

三、实验

我们小组主要尝试了两种模型在数据集 Beauty 和 Cell 上的表现，并依据表现结果来确定模型的选择。两种模型分别是是 2021 年发表的基于图神经网络的 GCE-GNN 和 2022 年发表的基于滤波器和前馈神经网络的 FMLP-Rec。

整体的建模思路为数据处理->模型训练与选择->模型调优

对于 pytorch 项目文件夹，在文件夹下的 readme.md 文件中有详细的项目结构和运行方法介绍，下面是技术处理细节：

（一）数据集的处理

因为大作业的原始数据集隐去了用户的 ID 和物品交互的时间戳，所以没有办法通过时间戳的先后来将数据集进行准确划分。通过对论文的研读和学习，我们发现在序列推荐中存在两种数据集的划分方式。

（1）第一种划分数据集的方式是将一个用户完整的序列作为一整条训练数据，然后某一部分的用户作为训练集，其余用户划分测试集（验证集）；

（2）第二种划分数据集的方式是将所有用户近期以前（例如时间戳倒数第二个以前）的部分作为训练数据，然后将完整的交互序列作为测试集（验证集）。

经过我们的分析和与老师的交流讨论，我们觉得第一种划分数据集的方式存在数据泄露的问题，例如用户 1 的交互数据已经包括了 2022 年 6 月份的数据，但是可能用户 2 的交互数据只包括了 2022 年 3 月份的数据，也就是存在用“未来的数据”去预测“过去的数据”，这显然是不合理的。所以我们采用了第二种划分数据集的方式。对原始数据集，我们将原本属于一个交互序列，但是经过切分子序列的所有序列合并成一条完整的训练数据。

（二）模型选择

我们分别采用 GCE-GNN 和 FMLP-Rec 两种模型，在没有调过参数的情况下，在 Beauty 和 Cell 上分别进行了测试，最终选择 GCE-GNN 模型。下面是实验的结果：

1. FMLP-Rec 模型在不同数据集上的表现

(1) 在 Amazon_Beauty 数据集上的表现:

FMLP-Rec	HR@5	MRR@5	HR@20	MRR@20
Beauty	9.37%	5.44%	14.15%	7.65%

(2) 在 Amazon_Cell 数据集上的表现:

FMLP-Rec	HR@5	MRR@5	HR@20	MRR@20
Cell	8.74%	4.32%	10.12%	5.96%

2. GCE-GNN 模型在不同数据集上的表现

(1) 在 Amazon_Beauty 数据集上的表现:

GCE-GNN	HR@5	MRR@5	HR@20	MRR@20
Beauty	9.57%	6.17%	18.01%	8.25%

(2) 在 Amazon_Cell 数据集上的表现:

GCE-GNN	HR@5	MRR@5	HR@20	MRR@20
Cell	8.98%	5.38%	14.37%	6.56%

(三) 模型优化

根据短序列推荐领域的常用数据和本数据集的特性,我们引入先验知识来进一步优化模型。首先,我们发现相同的 item id 不会连续出现,即不考虑用户反复点击同一种商品的情况;更进一步,我们认为在同一条完整的序列中任一 item id 只出现一次,即不考虑用户反复点击相同商品的情况。

根据以上信息,我们设计第一种专家系统(expertise.py)如下: 1. 使用训练集的所有信息构建字典,存储每一个 item id 后一位曾出现过的所有 id 2. 按照出现频次由大到小进行排序。3. 对于频次相同的 id,如果在深度学习模型给出的结果中出现过,按照

模型结果中的先后顺序进行排序。4. 将专家系统的结果链接到深度学习模型结果的前端形成新的预测 TOP20

第二种专家系统(expertise_2.py)如下：1. 使用训练集的所有信息构建字典，存储每一个 item id 后一位曾出现过的所有 id。 2. 按照出现频次由大到小进行排序。 3. 将深度学习模型结果中出现过的 id 删除。 4. 将专家系统的结果替换深度学习模型结果的末尾形成新的预测 TOP20。

根据实验和理论分析，我们最终决定使用第一种专家系统。

同时，我们在 GCE-GNN 模型的训练过程中也加入先验知识，修改了模型，在每个 epoch，无论是训练还是验证，我们将纯粹的 GCE-GNN 模型输出略作修改，去掉输出中包含输入 id 的部分形成新的输出，这个输出作为我们模型的最终输出。

（四）模型调参

我们通过网格搜索的方式对 GCE-GNN 的参数进行了调整，最终调整后的关键参数如下表所示：

（1）Amazon_Beauty 数据集上的参数：

batch_size	120
lr	0.005
lr_dc	0.2
dropout_gcn	0.4
dropout_local	0.5
dropout_global	0.3
alpha	0.2
seed	2022

（2）Amazon_Cell 数据集上的参数：

batch_size	120
lr	0.005
lr_dc	0.2
dropout_gcn	0.4
dropout_local	0.5
dropout_global	0.3
alpha	0.2
seed	2

（五）实验结果

经过参数调整以后的 GCE-GNN 在 Beauty 和 Cell 的验证集上的表现如下：

（1）在 Amazon_Beauty 数据集上的表现：

GCE-GNN	HR@5	MRR@5	HR@20	MRR@20
Beauty	14.88%	10.22%	22.96%	11.03%

（2）在 Amazon_Cell 数据集上的参数：

GCE-GNN	HR@5	MRR@5	HR@20	MRR@20
Cell	10.65%	6.86%	17.95%	7.58%

四、总结和展望

短序列推荐任务是一个应用十分火热的课题，在生活中随处可见，很有研究的价值。本次项目更加偏向于纯粹的商品数据关系挖掘，在只有 id 信息的情况下进行序列预测会更加依赖于算法对数据的挖掘。

我们在将近一个月的探索与尝试中，愈发感受到短序列推荐任务的奇妙和魅力。刚开始我们只是对其有大概模糊的了解，搜寻论文模型进行尝试，初步建立起对问题的基本认知。然后我们慢慢研读不同的论文和一些文献综述，学习了几类热门基本模型，结合本次数据集特征，最终决定了 MLP 和 GNN 两个模型来 battle 并对他们进行深入优化改良，得到最终结果。一路走下来，对问题一点一滴的研磨都是在深度学习领域的一点提升，也是自我知识体系的一次次完善，受益匪浅。

在其中我们也遇到了几个问题。我们发现即使设置了 numpy 和 torch 的随机种子，在不同的设备上运行相同参数的模型得到的结果也会有所差距，查阅文献后发现是 torch 和 cuda 内涵的设备随机数生成方法不同导致的，作为一个小插曲也让我们学到了代码更底层的知识，很有意思。

因为无法得知模型在 test 数据集上的结果，我们尝试横切 train 数据集，取出 train 中 5% 的会话数据作为“test_session”进行训练（GCE-GNN-master-try）。通过验证集与测试集上的结果对比我们发现，模型存在过拟合现象。但是无论 drop out 的比率调整为多少，或早停设置为几，或正则化如何设置，测试集的效果总是比验证集低至少一倍，这个问题困扰了我们很长一段时间。我们首先想到可能是因为数据划分方式导致一些 id 没有被训练到，导致验证集与测试集差距明显。后来我们想到，既然传统的防止过拟合方法遇到瓶颈，就需要在模型结构层面进行修改。结合 FMLP-Rec 模型的思想，我们可以在 GCE-GNN 模型中加入滤波器层，滤掉一些“杂音”，减轻过拟合现象，这是我们对于模型的未来展望之一。

第二个展望是在于数据层面，我们在调参过程中也同样发现，仅提供 id 信息的情况下，现在的深度学习模型预测效果仍然处于发展阶段，召回率等指标限制了它不能完全用于真正的企业实战中，这一个方向还有很多可以走出的创新。对于序列推荐的输入数据可以更加多元化，比如用户信息、时间信息，甚至是突发事件、热点事件对于接下来人们可能会搜索的内容的影响。我相信随着短序列推荐技术的迭代进步，越来越复杂的因素会被考虑，不仅仅停留在单纯的对于结构化数据的处理预测，还能结合比如自然语言处理等方面的成果来完善短序列推荐，最终发展为一个成熟的技术领域！