

SocialLGN: Light graph convolution network for social recommendation

摘要

图神经网络已被应用于推荐系统，来从用户-物品图中学习用户和物品的表示。在最新进的将图神经网络应用于社交推荐的方法中，主要有两个挑战：(i) 如何从用户-物品交互图和社交图中准确地学习用户和物品的表示，以及 (ii) 基于用户同时出现在这两个图中，如何整合这两个图中学习到新的用户表示。针对这些挑战，选择的文章提出了一种新的社交推荐系统 SocialLGN。在 SocialLGN 中，每个用户和物品的表示在用户-物品交互图中用浅图卷积层传播；同时，用户的表示在社交图中传播。在此基础上，还设计了一种图融合操作来聚合传播过程中的用户表示。加权和用于组合每一层学习的表示。原作者还在两个真实世界的数据集上进行了全面的实验，结果表明所提出的 SocialLGN 优于 SOTA 方法，尤其是在处理冷启动问题方面。

关键词：社交推荐；图卷积网络；推荐系统；嵌入传播

1 引言

随着用户行为数据的不断增长和多样化，如何有效地利用社交网络信息来提升推荐性能，已经成为一个重要的研究方向。传统的推荐系统大多依赖于用户与物品之间的交互信息，但这些方法往往无法充分利用用户之间的社交关系，尤其是在冷启动问题和稀疏数据问题上，存在明显的局限性。近年来，图卷积网络（GCN）由于其图结构数据上的强大表示能力，被广泛应用于推荐系统中。然而，现有的 GCN 模型往往只关注用户与物品之间的交互信息，忽略了社交网络中的用户间关系。

SocialLGN 正是应对这一挑战提出的一个创新性模型。该模型通过在用户-物品交互图和社交图中同时传播用户的表示，能够同时学习来自用户交互和社交关系的多层次特征。通过设计新的图融合操作，SocialLGN 将来自不同图谱的用户表示进行有效融合，进而提升推荐系统的准确性和泛化能力。此外，SocialLGN 还有效缓解了推荐系统中的冷启动问题，特别是在缺乏充分交互数据的新用户或新物品场景下，展现出了较为优越的表现。

从选题依据来看，社交推荐系统结合了社交网络信息和推荐系统的优势，能为个性化推荐提供更加丰富的上下文信息，具有较大的应用潜力。选题的意义在于，我们不仅能够验证其在现实数据集上的表现，还能探索如何更好地融合多源信息，提升推荐系统的精度和效率。

2 相关工作

此部分对课题内容相关的工作进行简要的分类概括与描述, 二级标题中的内容为示意, 可按照行文内容进行增删与更改, 若二级标题无法对描述内容进行概括, 可自行增加三级标题, 后面内容同样如此, 引文的 bib 文件统一粘贴到 `refs.bib` 中并采用如下引用方式 [16]。

2.1 社交推荐

随着社交平台的日益普及, 人们认识到可以利用社交信息来提高推荐系统的性能 [25, 26]。大多数社交推荐模型依赖于同质性 [20] 和社会影响 [3] 来解决推荐系统中常见的数据稀疏问题和冷启动问题。意识到矩阵分解 (MF) [13] 技术能够很好地捕捉社会关系的实现, 社交推荐系统的早期设计大多基于 MF。这些推荐系统通常可以根据它们利用社交信息的方式分为 3 个类别 [23]: 协同分解方法 [18]、集成方法 [17] 和正则化方法 [19]。

SBPR [29] 是一个成对贝叶斯个性化排名 (BPR) 模型, 该模型假设用户倾向于对朋友喜欢的物品给予更高的评级。它定义了一类称为“社会反馈”的新物品来表示社交网络中的信息。TrustSVD [7, 8] 被开发用于应用 SVD++ [12], 这是一种基于 MF 的模型, 已广泛用于社交推荐系统中。TrustSVD 利用社会关系, 将社会邻居的隐式反馈视为用户的辅助隐式反馈。ContextMF [10] 是一个集体基于 MF 的框架, 具有额外的正则化项, 其中正则化项旨在结合上下文信号和社交网络。TrustMF [26] 是一个基于 MF 的模型, 它将社会信任网络分解为两个低维潜在特征空间。通过这种方式, 它可以捕获用户间的互相影响。上述工作多基于 MF 技术, 但最近的研究开发了深度神经网络 [5] 和图表示 [14] 方法来表示用户和物品。这些方法利用图卷积网络的多层结构来更好地挖掘用户和物品之间的非线性关系。

2.2 图神经网络和图卷积网络在社交推荐中的应用

图神经网络的主要原理是迭代地从目标节点的相邻节点中获取再聚合信息, 得到图中每个节点的表示。一些文献将 GNN 应用于社交推荐问题。GraphRec [6] 是第一个通过融合用户-物品交互图中的单跳相邻节点信息和社交图来学习用户表示的研究。DiffNet [25] 将两个图中的用户表示相加得到最终的用户表示。对于用户-物品交互图, 它对单跳相邻节点执行平均池化操作; 对于社交图, 它使用 GNN 传播每个用户的表示。DiffNet++ [24] 是 DiffNet 的扩展。它通过设计一个多级图注意力网络来同时对用户对物品的兴趣和用户相互影响进行建模来提高原始 DiffNet 的性能。DiffNet 和 DiffNet++ 都使用逐层传播结构来描述社会影响的动态扩散过程。参考文献 [32] 提出了一种双图注意力网络来协作学习双重社会影响的表示。DGRec [22] 使用两个循环神经网络根据上下文对动态用户行为和社会影响进行建模。

GCN 是一种卷积网络, 它直接对图结构进行编码, 并通过在图中传播节点的特征来获得图中的节点表示。GCN 将传统上应用于具有欧几里得结构的数据的卷积操作扩展到具有非欧几里得结构的数据。近年来, GCNs 已应用于图表示学习 (例如 [11]), 并显示出令人满意的性能。早期的图卷积方法 [1, 4] 在谱域中定义, 并且计算成本很高。后来的研究 [9, 11] 重新定义了空间域中的图卷积方法, 以较低的计算成本获得了更好的可解释性; 这些方法现在已被广泛采用并应用于不同的领域, 如人体骨架建模 [21]、多标签图像识别 [2] 和文本分类 [15]。GCN 也被应用于社交推荐 [6, 24], 他们迭代地聚合目标节点相邻节点的嵌入, 以细化目标节点在表示用户-物品交互的图上的嵌入。Yu 等人 [27] 提出了一种基于 GCN 的深度对抗框架

来解决社交推荐中存在的三个问题。文献 [28] 提出了一种多通道超图卷积网络，该网络利用高阶用户关系来提高推荐的准确性。

3 本文方法

3.1 本文方法概述

该系统的整体架构如图 1 所示。该系统由三个主要部分组成:(i) 嵌入层，它从用户标识符(ids) 初始化用户和物品的嵌入;(ii) 多个传播层，在社交图和用户-物品交互图中传播嵌入;(iii) 预测层，根据用户和物品的最终嵌入表示预测一条连接用户和物品的边的值，即未来多大可能性用户会与该物品产生交互。

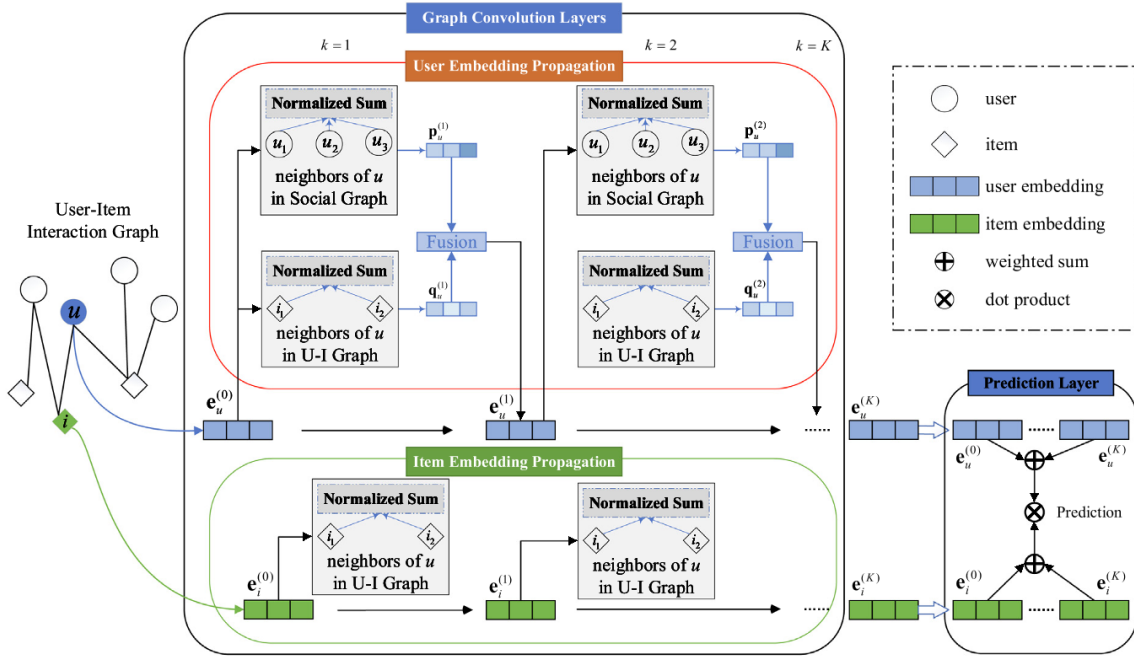


图 1. SocialLGN 架构设计

3.2 嵌入层

嵌入层用一个向量来表示一个用户，或者一个物品，且表示用户和物品的向量长度是一致的: $\mathbf{e} \in \mathbb{R}^d$ ，且这个长度是事先指定的。这个向量被称为用户或者物品的嵌入，它们的可以通过一些已知的特征来初始化，也可以遵循概率分布来进行随机初始化。这些嵌入会在后续的训练中被更新，使它们可以更好的表达网络中的原实体之间的关系。假设我们的网络中有 N 名用户和 M 件物品，它们的嵌入可以被表示成两个矩阵: $\mathbf{E}_U \in \mathbb{R}^{N \times d}$ (users, 用户) 和 $\mathbf{E}_I \in \mathbb{R}^{M \times d}$ (items, 物品)。

3.3 传播层

基于嵌入层生成的用户和物品的嵌入，使用一个更轻量的图卷积操作在社交图和交互图上更新嵌入表示。所谓更轻量，即在传统 GCN [11] 的基础上移除了线性变化层和非线性激

活。在第 k 个传播层中，每次都取第 $(k-1)$ 层的嵌入，对其进行更新，得到当前层的嵌入表达，其更新过程如下。

对于用户 u ，作者使用用户-物品交互图和社交图来更新从第 $(k-1)$ 层输出的用户嵌入（记为 $\mathbf{e}_u^{(k-1)}$ ）。为此，分别在用户-物品交互图和社交图中为用户 u 生成两个嵌入，记为 $\mathbf{q}_u^{(k)}$ 和 $\mathbf{p}_u^{(k)}$ ：

$$\mathbf{q}_u^{(k)} = \sum_{i \in \mathcal{N}_u^I} \frac{1}{\sqrt{|\mathcal{N}_u^I| \cdot |\mathcal{N}_i^I|}} \mathbf{e}_i^{(k-1)},$$

$$\mathbf{p}_u^{(k)} = \sum_{v \in \mathcal{N}_u^S} \frac{1}{\sqrt{|\mathcal{N}_u^S| \cdot |\mathcal{N}_v^S|}} \mathbf{e}_v^{(k-1)},$$

其中， \mathcal{N}_u^S 表示用户 u 在社交图中的邻居集合， \mathcal{N}_u^I 表示用户 u 在用户-物品交互图中的邻居集合。上面两个公式本质上通过聚合两个图中用户邻居节点的嵌入来更新用户 u 的嵌入。 $\mathbf{q}_u^{(k)}$ 和 $\mathbf{p}_u^{(k)}$ 随后被融合生成更新后的嵌入 $\mathbf{e}_u^{(k)}$ 。这种融合过程可以通用地表示为：

$$\mathbf{e}_u^{(k)} = f(\mathbf{q}_u^{(k)}, \mathbf{p}_u^{(k)}),$$

其中 $f(\cdot)$ 表示图融合操作。在本研究中，作者提出了以下图融合操作来生成 $\mathbf{e}_u^{(k)}$ ：

$$\tilde{\mathbf{e}}_u^{(k)} = \mathbf{W}_3 (\sigma(\mathbf{W}_1 \mathbf{p}_u^{(k)}) \parallel \sigma(\mathbf{W}_2 \mathbf{q}_u^{(k)})),$$

$$\mathbf{e}_u^{(k)} = \frac{\tilde{\mathbf{e}}_u^{(k)}}{\|\tilde{\mathbf{e}}_u^{(k)}\|_2},$$

其中 $\sigma(\cdot)$ 是 \tanh 激活函数， $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d \times d}$ ， $\mathbf{W}_3 \in \mathbb{R}^{d \times 2d}$ 是可训练矩阵。其中归一化操作，抑制了 $\mathbf{e}_u^{(k)}$ 随 k 增大的值增长。

对于物品 i 的第 $(k-1)$ 层嵌入 $\mathbf{e}_i^{(k-1)}$ ，其在用户-物品交互图中通过以下方式传播：

$$\mathbf{e}_i^{(k)} = \sum_{u \in \mathcal{N}_i^I} \frac{1}{\sqrt{|\mathcal{N}_i^I| \cdot |\mathcal{N}_u^I|}} \mathbf{e}_u^{(k-1)},$$

其中， \mathcal{N}_u^I 和 \mathcal{N}_i^I 分别表示用户 u 和物品 i 在用户-物品交互图中的邻居集合。

3.4 预测层

嵌入经过 K 层传播，我们将每一层的嵌入进行加权求和得到最终的嵌入表示。例如对于用户和物品，我们分别获去了它们在每一层传播后的嵌入结果： $\{\mathbf{e}_u^{(0)}, \dots, \mathbf{e}_u^{(K)}\}$ 和 $\{\mathbf{e}_i^{(0)}, \dots, \mathbf{e}_i^{(K)}\}$ 它们的最终嵌入可以写作：

$$\mathbf{e}_u = \sum_{k=0}^K \alpha_k \mathbf{e}_u^{(k)}$$

$$\mathbf{e}_i = \sum_{k=0}^K \alpha_k \mathbf{e}_i^{(k)}$$

最终计算用户 u 和物品 i 的得分是通过求两个向量的内积，直觉上可以理解为计算该用户和该物品的相关性：

$$\hat{y}_{ui} = \mathbf{e}_u^T \mathbf{e}_i$$

3.5 损失函数定义

该模型使用的损失函数是 Bayesian Personalized Ranking (BPR)。BPR 的优势在于，它的评估是区分用户的，即如果对于某个用户，用户应该对物品 A 更感兴趣，而不是物品 B，那么模型就应该给物品 A 更高的得分；如果存在另一名用户，也是同理，但不会去在用户间进行比较，这样的做法是更合理的。它的公式如下，这里和原来公式不同的是还引入了一个正则项：

$$\mathcal{L}_{BPR} = \sum_{(u,i) \in N_U^I \cup (u,j) \notin N_U^I} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|\mathbf{E}\|_2^2$$

4 复现细节

4.1 与已有开源代码对比

我的复现直接参考了原文提供的代码，现来自以下 GitHub 仓库：[SocialLGN](#)。该代码实现有几个不足：

- 除了训练模型常需要的超参数之外，作者没有提供例如训练多少个 epoch。
- 该代码保存模型参数的方式是，每隔几个 epoch 就保留一次，通常需要训练几千个 epochs，这样非常占用存储空间。
- 对于结果，在每个 epoch 训练完成后，直接在命令行输出，不便于观察和保存。
- 模型训练的时间比较长，常常会发生中断，但是对于中断后恢复上次中断结果的代码部分不完善，导致需要从头训练。

对于上面的没点，我都对其进行了改进：

- 只持久化两个版本的模型权重，一个是当前最好的模型权重（在验证集上的表现），另一个是最新的权重。
- 保存以上两个权重的时候，也保存 optimizer 的状态和 epoch 值，保存成一个 checkpoint 以支持状态恢复。
- 将结果写入 tensorboard，方便观察 loss 曲线和其他的指标，观察模型的训练过程是否正常，例如有没有过拟合。
- 支持从某个 checkpoint 恢复，继续上一次被中断的训练。

4.2 实验环境搭建

1. 创建 Conda 环境：使用以下命令创建 Python 3.8 的 Conda 环境：

```
conda create -n light-gcn python=3.8
```

2. 激活环境：

```
conda activate light-gcn
```

3. 安装必要的依赖：

- 深度学习框架：pytorch
- 图神经网络库：torch-geometric

因为作者没有提供每个依赖的具体版本，因此我在复现的过程中缺少什么包就安装哪些包；同时可能会遇到很多版本冲突，需要手动修复。

4. 为了方便复现实验环境，使用以下命令导出环境描述文件：

```
conda env export > environment.yml
```

生成的 environment.yml 文件内容如下：

```
name: light-gcn
channels:
  - defaults
dependencies:
  - _libgcc_mutex=0.1=main
  - _openmp_mutex=5.1=1_gnu
  - ca-certificates=2024.9.24=h06a4308_0
  - ld_impl_linux-64=2.40=h12ee557_0
  - libffi=3.4.4=h6a678d5_1
  - libgcc-ng=11.2.0=h1234567_1
  - libgomp=11.2.0=h1234567_1
  - libstdcxx-ng=11.2.0=h1234567_1
  - ncurses=6.4=h6a678d5_0
  - openssl=3.0.15=h5eee18b_0
  - pip=24.2=py38h06a4308_0
  - python=3.8.20=he870216_0
  - readline=8.2=h5eee18b_0
```

- setuptools=75.1.0=py38h06a4308_0
- sqlite=3.45.3=h5eee18b_0
- tk=8.6.14=h39e8969_0
- wheel=0.44.0=py38h06a4308_0
- xz=5.4.6=h5eee18b_1
- zlib=1.2.13=h5eee18b_1
- pip:
 - absl-py==2.1.0
 - astunparse==1.6.3
 - cachetools==5.5.0
 - certifi==2024.8.30
 - charset-normalizer==3.4.0
 - filelock==3.16.1
 - flatbuffers==24.3.25
 - fsspec==2024.10.0
 - gast==0.4.0
 - google-auth==2.36.0
 - google-auth-oauthlib==1.0.0
 - google-pasta==0.2.0
 - grpcio==1.68.0
 - h5py==3.11.0
 - idna==3.10
 - importlib-metadata==8.5.0
 - jinja2==3.1.4
 - joblib==1.4.2
 - keras==2.13.1
 - libclang==18.1.1
 - markdown==3.7
 - markupsafe==2.1.5
 - mpmath==1.3.0
 - networkx==3.1
 - numpy==1.24.3
 - nvidia-cublas-cu12==12.1.3.1
 - nvidia-cuda-cupti-cu12==12.1.105
 - nvidia-cuda-nvrtc-cu12==12.1.105
 - nvidia-cuda-runtime-cu12==12.1.105
 - nvidia-cudnn-cu12==9.1.0.70
 - nvidia-cufft-cu12==11.0.2.54
 - nvidia-curand-cu12==10.3.2.106
 - nvidia-cusolver-cu12==11.4.5.107

- nvidia-cusparse-cu12==12.1.0.106
- nvidia-nccl-cu12==2.20.5
- nvidia-nvjitlink-cu12==12.6.85
- nvidia-nvtx-cu12==12.1.105
- oauthlib==3.2.2
- opt-einsum==3.4.0
- packaging==24.2
- pandas==2.0.3
- protobuf==4.25.5
- pyasn1==0.6.1
- pyasn1-modules==0.4.1
- python-dateutil==2.9.0.post0
- pytz==2024.2
- requests==2.32.3
- requests-oauthlib==2.0.0
- rsa==4.9
- scikit-learn==1.3.2
- scipy==1.10.1
- six==1.16.0
- sympy==1.13.3
- tensorboard==2.13.0
- tensorboard-data-server==0.7.2
- tensorflow==2.13.1
- tensorflow-estimator==2.13.0
- tensorflow-io-gcs-filesystem==0.34.0
- termcolor==2.4.0
- threadpoolctl==3.5.0
- torch==2.4.1
- tqdm==4.67.1
- triton==3.0.0
- typing-extensions==4.5.0
- tzdata==2024.2
- urllib3==2.2.3
- werkzeug==3.0.6
- wrapt==1.17.0
- zipp==3.20.2

prefix: /home/data3/xiongekouqin/tool/miniconda3/envs/light-gcn

4.3 界面分析与使用说明

本文的实验没有实际的可视化界面，进入项目的根目录后可以使用以下脚本来运行：


```
python main.py
--model=SocialLGN
--dataset=lastfm
--decay=1e-4
--lr=0.001
--layer=3
--seed=2020
--topks="[10,20]"
--recdim=64
--bpr_batch=2048
```

4.4 创新点

复现的创新点主要有以下两个，但是由于实验非常耗费计算资源没有进行验证。

在社交图中使用更具表现力的 GNN 模块：当前的方法是对每个用户的邻居节点进行简单的聚合，即对邻居节点的特征进行平均或加权求和，这种方式无法有效区分不同邻居对目标用户的影响程度。提出的改进方法是引入 GAT 模块（Graph Attention Network），通过注意力机制为每个邻居节点分配不同的重要性权重。GAT 模块能够根据邻居节点的特征以及与目标节点的关系，动态地调整邻居的重要性，从而更精确地表示用户的社交关系和特性。

考虑新的方法来融合来自两个图的用户嵌入：当前的融合方式可能无法充分利用用户在用户-物品交互图和社交图中的结构信息以及特征信息。提出的改进方向是设计一种新的用户嵌入融合方法，将两个图中学习到的用户特征有效地结合起来，进一步提升模型的表达能力和推荐效果。这两个创新点旨在增强模型对用户行为的建模能力，通过更精细的图操作和特征融合，提高推荐系统在社交推荐任务中的性能表现。

5 实验结果分析

训练过程中的 loss 曲线如下，可以观察出 loss 曲线比较平滑，在持续在优化。

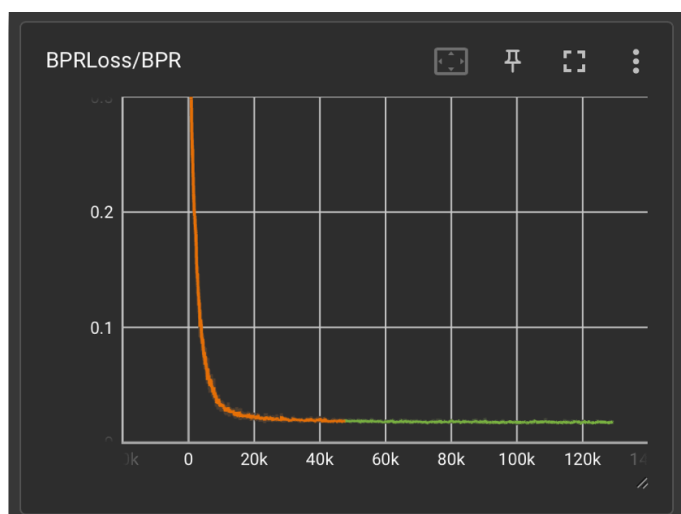


图 2. 实验结果: BPRLoss

除了训练过程中的 loss，还记录了在验证集上的指标。其中可以观察到在标准数据上，各项指标虽有波动但是总体呈持续提升趋势；而在冷启动数据（新加入的用户，暂时没有太多社交互动和物品的交互）中，指标的波动很大。这实际上是正常现象，因为训练是基于用户和其他用户或者商品的交互，对于这些冷启动数据，并没有充足的数据对其进行优化。但是尽管如此，其表现也优于当时的同类其他模型。

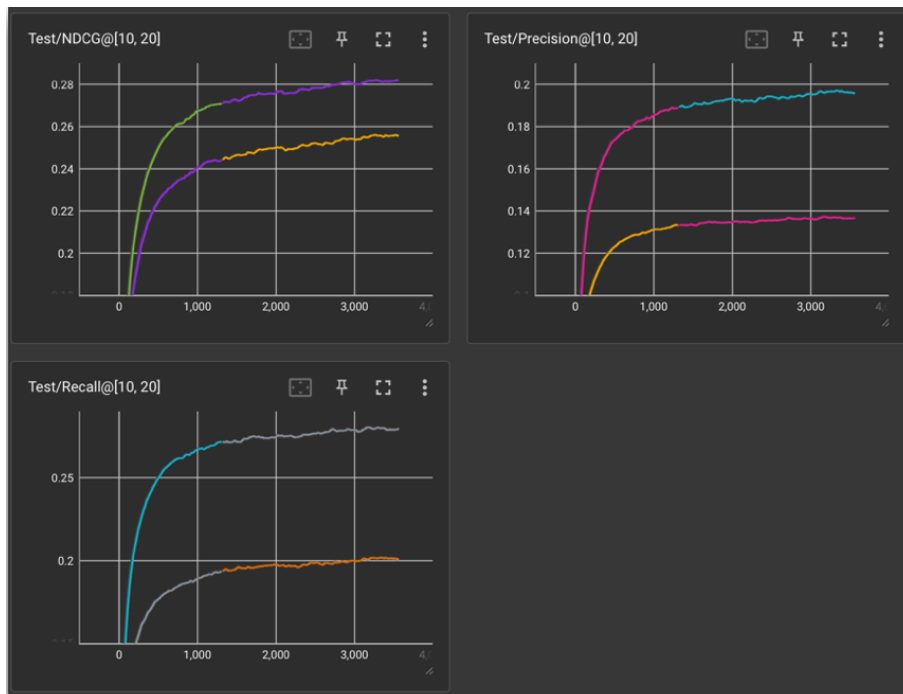


图 3. 实验结果: 验证集标准数据

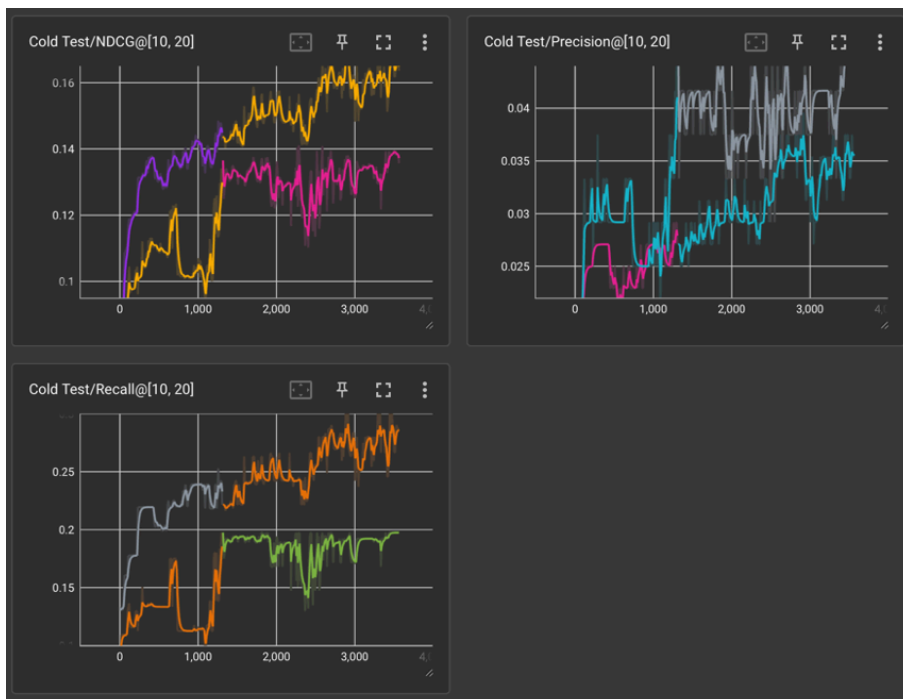


图 4. 实验结果: 验证集冷启动数据

就主要指标，复现结果和原文结果对比如下，可以观察到大部分指标都比较接近或者达

到了原文的结果，一定程度上反映了复现的正确性。

表 1. 实验结果对比

Metric	Author's	Mine
Recall@10	0.2026	0.2019
NDCG@10	0.2566	0.2465
Precision@10	0.1972	0.1968
(cold) Recall@10	0.1974	0.1974
(cold) NDCG@10	0.1419	0.1410
(cold) Precision@10	0.0458	0.0458

6 总结与展望

本文复现了 SocialLGN 模型，在复现报告中描述了使用的方法理论，基于原作者的开源代码进行了实验，与原文的结果进行了对比。结果显示，复现的结果与原文提供的结果接近。在复现的基础上，还在训练流程上进行了优化，使训练更方便结果也更容易可视化。原论文提供的代码没有清楚给出所需依赖的版本细节，本复现提供了完整的依赖版本以方便未来想要进行复现的使用者。最后，也提出了对模型进行进一步优化的建议。

尽管复现工作取得了一定的成果，但在具体实现过程中仍存在一些不足之处。首先，部分超参数的设置没有进行系统性的优化，可能限制了模型的性能上限。此外，由于硬件资源的限制，实验仅在中小规模的数据集上进行，未能验证模型在大规模数据集上的扩展性。最后，一些创新点的实现细节可能尚未完全复现。

未来的研究可以围绕以下方向展开：首先，对模型的超参数进行自动化调优，以进一步提升其性能。其次，可在更大规模的真实数据集上开展实验，验证模型的扩展性与鲁棒性。此外，可以尝试结合其他图神经网络的技术，如动态图建模或多模态信息融合，进一步改进模型的效果。最后，也可以结合 GNNExplainer 等解释性技术，对模型的预测结果进行可解释性分析，以提高模型的透明性和可信度。

参考文献

- [1] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *International Conference on Learning Representations, International Conference on Learning Representations*, May 2014.
- [2] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.
- [3] Robert B. Cialdini and Noah J. Goldstein. Social influence: Compliance and conformity. *Annual Review of Psychology*, page 591–621, Feb 2004.

- [4] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *arXiv: Learning, arXiv: Learning*, Jun 2016.
- [5] Wenqi Fan, Qing Li, and Min Cheng. Deep modeling of social relations for recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, Jun 2022.
- [6] Wenqi Fan, Yao Ma, Qing Li, Eric Zhao, Jiliang Tang, Yuan He, and Dawei Yin. Graph neural networks for social recommendation.
- [7] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. A novel recommendation model regularized with user trust and item ratings. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1607–1620, Jul 2016.
- [8] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. *Proceedings of the AAAI Conference on Artificial Intelligence*, Jun 2022.
- [9] WilliamL. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Neural Information Processing Systems, Neural Information Processing Systems*, Jun 2017.
- [10] Meng Jiang, Peng Cui, Fei Wang, Wenwu Zhu, and Shiqiang Yang. Scalable recommendation with social contextual information. *IEEE Transactions on Knowledge and Data Engineering*, 26(11):2789–2802, Nov 2014.
- [11] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv: Learning, arXiv: Learning*, Sep 2016.
- [12] Yehuda Koren. Factorization meets the neighborhood. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, Aug 2008.
- [13] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, page 30–37, Aug 2009.
- [14] Chun-Yi Liu, Chuan Zhou, Jia Wu, Yue Hu, and Li Guo. Social recommendation with an essential preference space. *Proceedings of the AAAI Conference on Artificial Intelligence*, Jun 2022.
- [15] Xien Liu, Xinxin You, Xiao Zhang, Ji Wu, and Ping Lv. Tensor graph convolutional networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, page 8409–8416, Jun 2020.
- [16] Yilin Liu, Jiale Chen, Shanshan Pan, Daniel Cohen-Or, Hao Zhang, and Hui Huang. Split-and-Fit: Learning B-Reps via Structure-aware Voronoi Partitioning. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 43(4):108:1–108:13, 2024.

- [17] Hao Ma, Irwin King, and Michael R. Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, Jul 2009.
- [18] Hao Ma, Haixuan Yang, MichaelR Lyu, and Irwin King. Sorec: Social recommendation using probabilistic matrix factorization.
- [19] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, Feb 2011.
- [20] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, page 415–444, Aug 2001.
- [21] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.
- [22] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, Jan 2019.
- [23] Jiliang Tang, Xia Hu, and Huan Liu. Social recommendation: a review. *Social Network Analysis and Mining*, page 1113–1133, Dec 2013.
- [24] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE Transactions on Knowledge and Data Engineering*, page 4753–4766, Oct 2022.
- [25] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul 2019.
- [26] Bo Yang, Yu Lei, Jiming Liu, and Wenjie Li. Social collaborative filtering by trust. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1633–1647, Aug 2017.
- [27] Junliang Yu, Hongzhi Yin, Jundong Li, Min Gao, Zi Huang, and Lizhen Cui. Enhancing social recommendation with adversarial graph convolutional networks. *Cornell University - arXiv, Cornell University - arXiv*, Apr 2020.
- [28] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *Proceedings of the Web Conference 2021*, Apr 2021.

- [29] Tong Zhao, Julian McAuley, and Irwin King. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, Nov 2014.