计 算 机 研 究 与 发 展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202220643
61(4): 877−888, 2024

# Light-HGNN: 用于圈层内容推荐的轻量同质超图神经网络

李 挺[1] 金福生[1] 李荣华[1] 王国仁[1] 段焕中[2] 路彦雄[2]
[1](北京理工大学计算机学院 北京 100081)
[2](腾讯科技(北京)有限公司 北京 100080)
(litingbit@163.com)

# Light-HGNN: Lightweight Homogeneous Hypergraph Neural Network for Circle Content Recommendation

Li Ting[1], Jin Fusheng[1], Li Ronghua[1], Wang Guoren[1], Duan Huanzhong[2], and Lu Yanxiong[2]
[1](*School of Computer Science and Technology*, *Beijing Institute of Technology*, *Beijing* 100081)
[2](*Tencent Technology* (*Beijing*) *Co.*, *Ltd.*, *Beijing* 100080)

**Abstract** Graph neural network and hypergraph neural network (HGNN) have become research hotspots in the field of collaborative filtering recommendation. However, the interaction between users and projects in actual scenarios is very complex. As a result, there are high-order complex relationships among users, while ordinary graph structures can only express simple paired relationships. Stacking network structures easily leads to excessive smoothness of middle-tier representations, and the ability of user modeling, user similarity discovery and mining in sparse scenarios is weak. At the same time, the complex structure of heterogeneous hypergraph neural network makes the training efficiency of the model low. In the highly sparse data scene represented by WeChat "Search and Search" and other content platforms, the existing models have poor recommendation effect and weak interpretability of user representation for the circle content recommendation task based on the portrait of the user's group. Therefore, for this kind of task, a new lightweight homogeneous hypergraph neural network model is proposed, which includes three parts: the transformation of user interaction data into hypergraph, the generation of user representation sequence by convolution, and the calculation and filtering of user representation. Firstly, the user-item interaction data is transformed into a homogeneous hypergraph with only user nodes, and the initial value of user representation decoupling sequence is calculated. Then information propagation and sequence values are iteratively generated according to the Laplacian filter matrix of hypergraph, the model structure is simplified by convolution method without activation layer, and the weight matrix is generated for each sequence value according to the proposed mean difference JK attention mechanism. Finally, the coding of user representation is realized by weighted summation and filtering of decoupled sequences, and experiments on real data sets verify the relatively better effect of the proposed model.

**Key words** homogeneous hypergraph; hypergraph neural network; personalized recommendation; circle content recommendation; recommendation algorithm

**摘 要** 图神经网络和超图神经网络(hypergraph neural network, HGNN)已经成为协同过滤推荐领域的研

**收稿日期**: 2022-07-24; **修回日期**: 2022-05-19
**基金项目**: 国家重点研发计划项目(2020AAA0108503); 国家自然科学基金项目(62272045)
This work was supported by the National Key Research and Development Program of China(2020AAA0108503) and the National Natural Science Foundation of China (62272045).
**通信作者**: 金福生(jfs21cn@bit.edu.cn)

究热点.然而实际场景中用户和项目的交互非常复杂,导致用户之间存在高阶的复杂关系,而普通图结构只能表达简单的成对关系,对网络结构的堆叠容易导致中间层表征的过度平滑,在稀疏场景下的用户建模、用户相似性发现与挖掘方面能力较弱;同时,异质超图神经网络的复杂结构使得模型的训练效率较低.在以微信“搜一搜”等内容平台为代表的高度稀疏数据场景中,对于基于用户所属群体画像的圈层内容推荐任务,现有模型推荐效果差、用户表示的可解释性弱.因此,针对该类任务,提出了一个新的轻量同质超图神经网络模型,该模型包含用户交互数据至超图的转化、卷积生成用户表征序列、用户表征计算过滤.模型首先将用户-项目交互数据转化为只含用户节点的同质超图并计算得到用户表征解耦序列初始值,随后根据超图拉普拉斯过滤矩阵进行信息传播与序列值的迭代生成,通过不使用激活层的卷积方法简化模型结构,并根据提出的均值差 JK 注意力机制为每个序列值生成权重矩阵.最终,通过对解耦序列加权求和、过滤实现对用户表示的编码,并在真实数据集上进行实验验证了所提模型的相对更优效果.

**关键词** 同质超图;超图神经网络;个性化推荐;圈层内容推荐;推荐算法

**中图法分类号** TP311

社交网络平台的繁荣为社会的生产生活提供了相当程度的便利,极大地改变了人们的生活方式和行为方式.如今,互联网上的大多数创作都是由平台上的用户发布和消费的,一个更高效、更个性化的推荐算法可以更能理解用户的意图,帮助用户在平台上找到感兴趣的内容,缩短内容生产者和消费者之间的距离,提高平台的用户粘性[1].得益于用户数据量的迅速增长,通过平台内的画像研究可以发现,偏好相似的用户更有可能形成一个相互关联的圈层,即根据特定条件或方法聚集得到的用户群体,并在圈层中倾向分享彼此的交互,从而形成自发的协同过滤推荐.这是微博和推特等公共社交网络平台中的常见现象.基于这样的社会学发现[2],在实践中,不仅要考虑用户的偏好,还要关注他们的“朋友”,即存在社交关系的用户的交互数据,以改善推荐的效果.

现有的根据关系数据进行协同过滤的思路随着图神经网络理论的发展得到了深入研究,基于普通图的堆叠方法在一定程度上可以捕获多跳的社交关系[3],相关的研究已经得到了一些优秀的结果.但在实际场景中,社交关系与用户兴趣分布的不相关性给用户表征的学习带入噪声,同时用户和项目的交互非常复杂,用户之间也存在着更高阶的复杂关系信息,这种信息无法依靠简单成对关系建模.图卷积网络层的过度堆叠容易导致中间层表征的过度平滑[4],从而在稀疏场景下的用户建模、用户相似性发现与挖掘方面能力较弱.也有些工作提出使用超图对用户之间、用户与项目之间的关系进行高维建模,但异质超图神经网络(hypergraph neural network, HGNN)多选择将用户和项目等多种不同属性的节点统一到同一个图中进行模式挖掘[5-8],其中的复杂结构增加了超图卷积操作的复杂度,并降低了模型的训练效率.另一方面,由于平台内容的快速增长和用户对平台的使用习惯不同,以腾讯微信“搜一搜”等内容平台为代表的应用产生了极大的用户活跃度和高度稀疏的生态数据分布.上述高度稀疏场景与传统场景有所不同,用户频繁交互形成的交互对象集合的重叠度极低(即交互呈现“孤岛”型分布),这提高了从交互拓扑中挖掘用户兴趣关联性的难度,对生态内的推荐算法提出了挑战,即如何完成高度稀疏数据场景下的个性化推荐任务并保证推荐效果和用户表示的可解释性.在场景中,面对根据用户所属群体的画像进行内容推荐的圈层内容推荐任务,单纯依靠拓扑关系进行图学习的现有模型很难达到令人满意的效果,且这些模型输出的用户与项目表征在可解释性上存在短板,难以胜任群体行为模式挖掘的相关任务.

出于解决高度稀疏数据场景下的圈层内容推荐和提高用户表征可解释性的任务考虑,针对圈层内容推荐任务,本文提出了一个新的轻量同质超图神经网络(lightweight homogeneous hypergraph neural network, Light-HGNN)模型,模型以项目为核心来构建连接用户节点的超边,从而构造用户节点同质超图来直接建模用户拓扑关系,降低了模型复杂性并缩小了参数规模,提升了训练的效率.如图 1 所示,模型忽略用户间的关系,并根据用户的交互历史数据提取蕴含在项目中的用户的兴趣信息,并结合超图拓扑关系和项目内容表示进行编码、输出用户在表示空间的表征,改善了表示的可解释性.

本文模型通过用户交互数据至超图的转化、卷积生成用户表征序列、用户表征计算与过滤来实现用户表示的编码.模型首先将用户-项目交互数据转化为只含用户节点的同质超图,并计算得到用户表

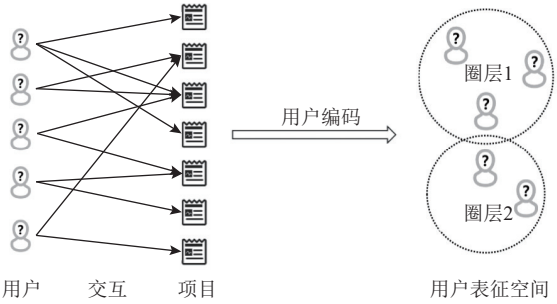



Fig. 1 Illustration of transferring interactions into circles of users

图 1 用户交互转化为用户圈的示意图

征解耦序列初始值，随后根据超图拉普拉斯过滤矩阵进行信息传播与序列值的迭代生成，通过不使用激活层的卷积方法简化模型结构，并根据提出的均值差跳跃知识（jump knowledge, JK）注意力机制为每个序列值生成权重矩阵. 最终，通过对解耦序列的加权求和、过滤完成对用户表示的编码. 在训练阶段，本文模型通过计算采样的贝叶斯个性化排名[9]损失来优化参数，通过相关测试指标的结果来确定、比较模型的效果.

本文的主要贡献有 4 点：

1）提出了一种将传统的用户-项目二元关系转化为同质超图的新方法来对用户数据进行建模. 方法根据用户表征的聚类来迭代地构建同质超图，从而生成用户表征序列，使构建的模型更加轻量、复杂性更低.

2）提出了根据交互对象的内容表示构建用户的表征的方法，从而提高表征的可解释性.

3）设计了新的更轻量的均值差 JK 注意力机制来生成用户表征序列对应的权重矩阵序列，并根据这一注意力机制完成高度稀疏数据条件下的圈层内容推荐任务中的编码任务.

4）在真实数据集上进行了模型对比实验，证明了模型相对更好的效果.

## 1 相关工作

本节主要从超图神经网络、个性化推荐以及协同过滤的相关方法与模型 3 个方面回顾相关领域的研究现状.

### 1.1 超图神经网络

超图中的超边可以同时连接多个节点，比图结构中的普通成对关系的表达能力更强，具备了对数据的高维属性进行建模的能力. 此外，图和超图之间结构的相似性使得图神经网络领域的现有方法更容易通过一些自适应调整后在超图上应用.

一些现有的工作已经证明了将图领域方法应用到超图领域的可行性与有效性. Feng 等人[10]在 2019 年发表超图神经网络研究论文，在超图中实现了类似图卷积网络[11]（graph convolutional network, GCN）的操作方法. Zhang 等人[12]发表了模型 Hyper-SAGNN，这个模型具有灵活处理多种形式超图的能力. 在后续的研究中，许多超图神经网络模型都开始借鉴其他网络模型中使用的机制，如 Bai 等人[13]在研究中提出的超图卷积注意力机制，Hu 等人[14]使用拉普拉斯平滑过滤器替代冗余的权重矩阵进行嵌入表示的迭代等. 这种尝试在各自针对的任务中表现出极好的适应性.

### 1.2 个性化推荐

个性化推荐的效果部分依赖于模型通过用户偏好挖掘得到的表示的质量. 最初的尝试主要集中在分析用户的查询、点击和参与主题等记录的工作上来提取用户的偏好，并得到用户浅层兴趣和深层兴趣的合适表示. Teevan 等人[15]以及 Dou 等人[16]分别在各自的工作中提出了衡量文档得分的方法，即分析用户历史操作中的点击来衡量文档的点击率. 在主题的研究领域，手动设计特征的巨大研究成本促使相关研究探索自动学习文档中涉及的主题表示的方法，根据相关分析[17]，许多使用排序学习方法处理组合特征的方法的效果比传统方法表现得更好.

随着推荐系统中数据量的快速增长，图与关系数据已经成为当前实践中数据源的重要组成. 深度学习的发展启发了许多研究者在图表示学习任务中采用神经网络方法并取得了显著成效. Wang 等人[18]提出的神经图协同过滤（neural graph collaborative filtering, NGCF）是较为经典的编码模型之一，可以在图结构中学习嵌入表示，显式地将协同过滤信号放入到嵌入过程中. He 等人[19]完成对 NGCF 模型结构解析的工作后提出了 LightGCN 模型，通过简化卷积层的设计来提高其对推荐任务的适用性，得到了优于 NGCF 的结果. 同时，基于对用户社交关系的研究也有些令人印象深刻的成果. Wang 等人[3]提出了顺序组推荐问题，并设计通过“组”这种结构来综合学习长短期用户表示. Zhou 等人[1]提出通过朋友关系网和相似搜索行为构建朋友圈，来解决用户搜索历史不足的问题.

随着研究的深入，部分研究关注到了图模型表达能力的局限，将关注点延伸至超图. Ai 等人[20]提出

的模型通过在不同用户和商品的知识以及静态和动态的关系的联合建模来构建知识图谱. Liu 等人[21]利用从用户-查询-商品中学习的逻辑结构表示和在逻辑路径上的不同实体之间的协作信号来改进个性化商品搜索. Xia 等人[6]提出了一种双通道超图卷积网络, 将会话数据建模为超图, 并将自监督学习融入到网络训练中. Yu 等人[2]通过研究超图结构上的用户间和用户与项目间的拓扑关系构建了自监督多通道卷积网络. 后续研究中 Wu 等人[7]将自监督对比学习任务和推荐系统的监督学习任务相结合, 提出了一种应用于用户-物品二分图推荐系统的图自监督学习框架.

但在很多情况下, 社会关系的复杂性意味着社交关系与兴趣分布不一定正相关. 推荐算法研究的另一个有效方向是进行元路径, 即交替的实体和关系序列的研究. 有些方法遵循这种思路研究了不同任务场景下异质图的元路径模式以挖掘用户潜在的搜索意图, 这些工作有助于提高推荐结果的可解释性. Hu 等人[22]提出基于相关任务上下文的元路径的方法, 通过使用协作注意力机制来生成基于特定交互的推荐结果. 但元路径模式的设计需要大量的经验, 且任务间多不能通用, 如何控制元路径的长度来平衡推荐效果和计算资源成本也是制约效果的瓶颈.

### 1.3 协同过滤算法

在不同研究思路的协同过滤算法中, 基于模型的协同过滤算法通过已知的用户和项目之间的交互来形成模型, 以便推荐系统能够识别具有更高复杂度的模式, 并在用户表示的帮助下获得推荐. 在该研究思路上有聚类方法[23]、贝叶斯网络[24]、关联算法[25]以及奇异值分解(SVD)[26]等方法. 在这些方法中, 矩阵分解(MF)[27]将所有用户和物品投影到一个共享的潜在因子空间, 并根据表示的内积计算结果进行预测. 算法因其性能和高扩展性曾被认为是最有效的协同过滤方法之一. 此外还有 Rendle 等人[9]提出的使用成对 log-sigmoid 函数进行优化的贝叶斯个性化排名(Bayesian personalized ranking, BPR)也是较为经典的算法框架; He 等人[28]提出了 Neural MF, 使用多层感知机来学习交互.

当前也有很多研究的注意力集中在用户和项目进行交互所生成的图结构中. 一些工作已经产生了很好的结果, 例如 SimRank[29], 当被相似的对象引用时, 将 2 个项目视为相似的对. 同时其他方法如 ItemRank[30]和 BiRank[31]则是受到标签传播机制的启发. 图卷积层的引入为协同过滤任务创造了另一个可行的路线. GCMC[32]将其应用于用户和项目之间的交互模型. 而在另一些模型中, PinSage[33]参考了 GraphSAGE[34]的思路, 将卷积层结构嵌入到商业推荐系统中. 而 Ji 等人[8]提出的 DHCF 模型尝试使用基于高阶连通性的跳跃超图卷积(jump hypergraph convolution, JHConv)方法将用户和项目表征进行同步优化, 对许多现有研究都具有启发性.

## 2 方法介绍

本节将深入地介绍 Light-HGNN 模型的结构, 并基于细节具体解释如何构建模型、优化模型并将模型应用到圈层内容推荐任务中.

图 2 总体展示了 Light-HGNN 的结构. 模型第 1 步根据用户和项目的关联关系将原始交互数据转化为超图, 并结合项目表示矩阵完成对用户表示和用户表示序列的初始化. 在第 2 步生成用户表示序列的过程中, 模型通过“用户聚类、同类用户交互扩展、构建超图、超图卷积”的方式迭代地生成用户表示序列, 每一个轮次的超图都是根据上一个轮次生成. 第 3 步, 整个序列经过注意力机制生成对应的权重序列. 在第 4 步对解耦序列加权求和、过滤得到最终的用户表示矩阵. 在整个流程中, 模型根据用户的聚类进行基于用户表示相似性的交互扩展, 并使用超图卷积方法加强被相同项目连接的用户间的信息流动. 这样将属于相同圈层用户的交互内容进行“共享”, 来达到挖掘用户的显式与隐式偏好、进行协同过滤的目的.

### 2.1 定 义

**定义 1.** 圈层. 圈层指根据某些条件或方法聚集起来的用户群体或用户集合. 用户可以同时属于根据不同划分确定的不同圈层来进行不同任务.

**定义 2.** 朋友圈与兴趣圈. 朋友圈指根据用户之间的社交关系(图 3 中实线关系, 如微信中的好友关系、微博等社交网络的关注关系等)构成的同质图结构, 图可以不是完全图或连通图. 兴趣圈是根据用户在兴趣表征空间中的分布情况构成的节点聚类(图 3 中圈), 在每一个聚类中, 用户的兴趣表示具有一定的相似性. 如图 3 所示.

**定义 3.** 圈层内容推荐. 圈层内容推荐任务属于个性化推荐任务, 在圈层内容推荐任务中, 模型只学习用户交互的项目的内容数据, 不考虑用户间关系, 输出带有圈层性的用户表征, 即兴趣表示相似的用

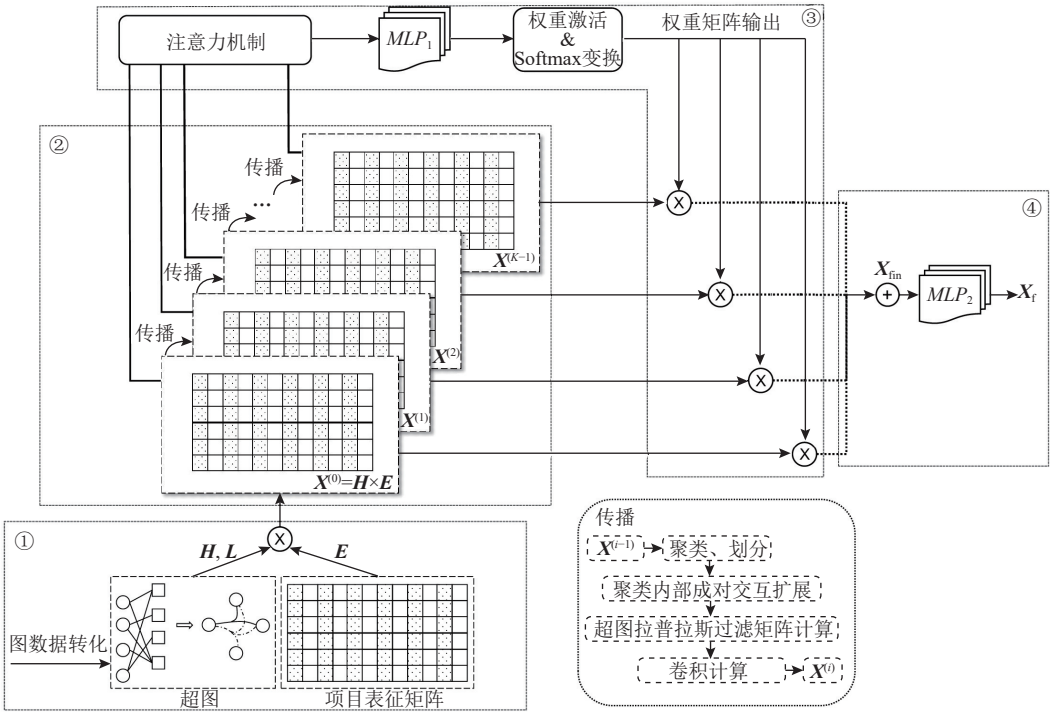



Fig. 2 Illustration of Light-HGNN

图 2 Light-HGNN 示意图

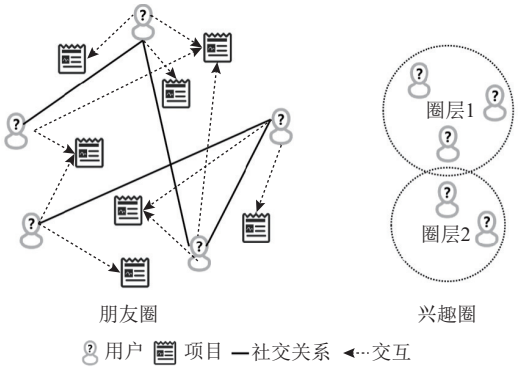



Fig. 3 Illustration of friend circle and interest circle

图 3 朋友圈与兴趣圈示意图

户在空间中聚集. 同时, 模型需要根据用户的数据输出用户的表示, 对于给定的候选项序列, 模型需要根据用户的表示和候选项的表示来计算候选项的分数, 该分数在某种程度上预测了用户对项目的兴趣程度, 或与用户可能和项目发生交互的概率正相关, 从而根据分数对候选项进行排序并生成推荐.

圈层内容推荐任务可以用公式化语言进一步描述. 对给定项目表征矩阵 $\boldsymbol{E}$ 和给定交互数据集 *Dataset*, 其中交互指用户和项目有交互, 例如, 用户阅读某文档是一次交互.*Dataset* 的定义如式(1)所示.

$$Dataset=\{(u,d)|u\in Set_{\text{user}}\wedge d\in Set_{\text{item}}\},\tag{1}$$

其中 $Set_{\text{user}}$, $Set_{\text{item}}$ 表示用户集合与项目集合, 模型 $M$ 应输出用户集合的表示矩阵 $\boldsymbol{U}$, 并根据用户 $u$ 的表示和给定候选项集 $D=\{D_i|\,0<i\leqslant n\}$ 对应输出候选项集的评分集 $P=\{P_i|\,0<i\leqslant n\}$, 其中 $P_i=M(u,D_i)$.

**定义 4.** 超图. 一般地, 一个超图可以被定义为 $G=(V,\xi,\boldsymbol{X},\boldsymbol{W})$, 其中 $V$ 为节点集, $\xi$ 为超边集, $\boldsymbol{X}\in\mathbb{R}^{|V|\times F}$ 为节点表示矩阵, $F$ 为节点表征向量的维度, $\boldsymbol{W}$ 为超边的权重矩阵. 超图中的节点与超边的连接情况可以用关联矩阵 $\boldsymbol{H}$ 来描述, 一般来说, 每个位置为 0 或 1, 如式(2)所示.

$$\boldsymbol{H}_u^d=\begin{cases}1, & v_u\in e_d,\\ 0, & \text{其他},\end{cases}\tag{2}$$

其中 $v_u\in e_d$ 表示超边 $d$ 连接节点 $u$. 根据超图关联矩阵, 可以进一步计算节点和超边的度矩阵.

### 2.2 超图建模与表征序列生成

#### 2.2.1 超图建模

1) 交互数据转化为超图. 最终的用户表示是由用户表征序列根据对应权重加权求和并过滤得到的, 而用户表征序列是由初始值通过迭代超图建模后计算得来的. 如式(1)所示的原始交互数据集 *Dataset* 中的数据, 可对应地构建用户和项目的关联矩阵 $\boldsymbol{H}^{\text{ori}}$. 其具体步骤为: 将矩阵初始化为 $|Set_{\text{user}}|\times|Set_{\text{item}}|$ 维度的零矩阵, 对任意交互记录 $(u,d)$, 对 $\boldsymbol{H}^{\text{ori}}$ 对应位置赋值使 $\boldsymbol{H}^{\text{ori}}$ 对应位置的值为 1. 此时可以得到原始交互数据转化来的关联矩阵. 如果将每一个项目视作超边

的核心,即以项目作为超边连接所有与之发生交互的用户,则可以将原始数据转化为一个超图,如图 4 所示,用户和项目的交互数据经处理对应构成了同质超图. 以用户 $A, C, D$ 与项目 $a$ 发生的交互为例,在超图中的用户 $A, C, D$ 将被项目 $a$ 转化得到超边连接. 原始数据转化得到的超图视作原始超图,$\boldsymbol{H}^{\text{ori}}$ 是它的关联矩阵.

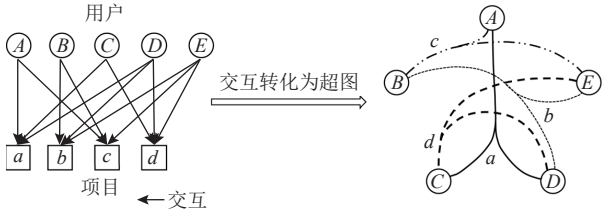



Fig. 4 Illustration of transferring interactions data into hypergraph

图 4 交互数据转化超图示意图

2) 用户表征序列初始化. 获得超图关联矩阵 $\boldsymbol{H}^{\text{ori}}$ 后,可以根据交互情况计算此时用户表征矩阵. 用户 $u$ 初始表示向量 $X_u^{(0)}$ 可以由交互项目的表示向量 $E_d$ 相加得到. 对用户 $u$ 的初始表示向量,如式(3)所示.

$$\boldsymbol{X}_u^{(0)} = \sum_{(u,d)\in Dataset} \boldsymbol{E}_d \,, \tag{3}$$

其中 $\boldsymbol{E}_d$ 表示项目 $d$ 的表示向量. 项目的向量表示可以来自上游任务,对于文档、视频等内容可以是关键词的多热或独热向量,亦或采集标题后使用自然语言处理模型输出,如词向量相加, Sentence2Vec 或 TF-IDF 以及其他广泛接受的语料库.

对整个用户集和项目集,可以进行矩阵尺度的计算,如式(4)所示.

$$\boldsymbol{X}^{(0)} = \boldsymbol{H}^{\text{ori}} \times \boldsymbol{E}, \tag{4}$$

其中 $\boldsymbol{E}$ 表示项目表征矩阵,则用户表征序列可以初始化为 $series=\{\boldsymbol{X}^{(0)}\}$. 从上述内容中可以得出,本文提出的模型以用户“交互”的对象的表示为基础进行用户编码,从而将用户与项目映射至相同的表征空间. 这里的“交互”包含实际发生的交互以及扩展得到的可能交互.

### 2.2.2 表征序列生成

得到初始化的用户表征序列后,可以根据表示序列预设长度 $K$ 和当前最新的用户表示进行超图建模从而迭代地生成完整用户表示序列. 其具体做法是:首先对上一轮的用户的表示 $\boldsymbol{X}^{(i-1)}(1 \leqslant i \leqslant K-1)$ 中的向量进行聚类,并根据聚类标签进行用户划分. 随后,在每个用户划分内部,对任意成对的用户进行交互扩展,即对每个用户对,在保留自身交互记录的同时,将自己的交互记录“分享”给对方,作为对方的扩展交互记录. 由于扩展的记录并非是用户的实际交互,所以在生成交互数据时需要记录的是形如(用户 $u$,项目 $d$,置信度 $c$)的三元组,其中置信度为扩展时 2 个用户表示之间的相似度,采用 2 个用户表示的余弦值并只取值在{0,1}(负值取 0,原始交互数据中的置信度默认为 1). 为了减少可能带入的噪音,可以在扩展时设置相似度阈值 $gate$ 来减少低相似度的扩展,扩展示意如图 5 所示,以 $A$ 和 $B$ 构成的聚类为例,其中 $(A, b)$ 与 $(B, a)$ 的交互记录是根据 2 个用户的表征相似性交互扩展得到的. 经过这样的处理就可以将式(1)中所示的原始交互数据 $Dataset$ 进行扩展,得到式(5).

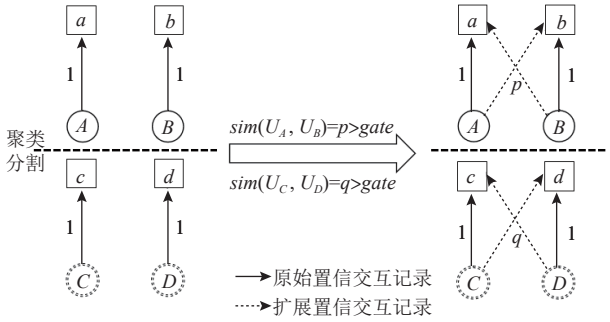



Fig. 5 Illustration of extension of interactions

图 5 交互扩展示意图

$$\begin{aligned} Dataset_{\text{extend}} = \{(u,d,c) | u \in Set_{\text{user}} \wedge \\ d \in Set_{\text{item}} \wedge c \in [0,1]\} \,. \end{aligned} \tag{5}$$

关联矩阵的生成和初始值生成类似,此时将置信度赋值到对应位置. 对任意交互记录 $(u, d, c)$,将 $\boldsymbol{H}^{(i-1)}$ 对应位置赋值使得 $\boldsymbol{H}_{u,d}^{(i-1)} = c$. 获取上一轮的经扩展后的超图邻接矩阵 $\boldsymbol{H}^{(i-1)}$,则可以根据它计算节点度矩阵 $\boldsymbol{D}_{\text{V}}^{(i-1)}$ 和超边度矩阵 $\boldsymbol{D}_{\text{E}}^{(i-1)}$,对上一轮的矩阵表示进行卷积后可以得到

$$\begin{aligned} \boldsymbol{X}^{(i)} = &(1-\gamma) \times \boldsymbol{X}^{(i-1)} + \gamma \times [\boldsymbol{D}_{\text{V}}^{(i-1)}]^{-1/2} \times \boldsymbol{H}^{(i-1)} \times \\ &\boldsymbol{W} \times [\boldsymbol{D}_{\text{E}}^{(i-1)}]^{-1} \times [\boldsymbol{H}^{(i-1)}]^{\text{T}} \times [\boldsymbol{D}_{\text{V}}^{(i-1)}]^{-1/2} \times \boldsymbol{X}^{(i-1)}, \end{aligned} \tag{6}$$

其中 $\gamma$ 为取值在 [0,1] 的超参数,$\boldsymbol{W}$ 可以设为相对应维度的单位矩阵. $\gamma$ 在式(6)中起到了限制信息流入规模的作用. 通过不使用激活层的卷积方法可以对表示矩阵进行快速迭代,实现模型结构的简化与轻量化.

根据文献 [14] 可以对式(6)进行化简,得到

$$\begin{aligned} \boldsymbol{X}^{(i)} = &\boldsymbol{X}^{(i-1)} - \gamma \times (\boldsymbol{I} - [\boldsymbol{D}_{\text{V}}^{(i-1)}]^{-1/2} \times \boldsymbol{H}^{(i-1)} \times \boldsymbol{W} \times \\ &[\boldsymbol{D}_{\text{E}}^{(i-1)}]^{-1} \times [\boldsymbol{H}^{(i-1)}]^{\text{T}} \times [\boldsymbol{D}_{\text{V}}^{(i-1)}]^{-1/2}) \times \boldsymbol{X}^{(i-1)}, \end{aligned} \tag{7}$$

其中 $\boldsymbol{I}$ 是单位矩阵. 取超图拉普拉斯过滤矩阵

$$\boldsymbol{L}^{(i-1)}=\boldsymbol{I}-[\boldsymbol{D}_{\mathrm{V}}^{(i-1)}]^{-1/2}\times\boldsymbol{H}^{(i-1)}\times\boldsymbol{W}\times [\boldsymbol{D}_{\mathrm{E}}^{(i-1)}]^{-1}\times[\boldsymbol{H}^{(i-1)}]^{\mathrm{T}}\times[\boldsymbol{D}_{\mathrm{V}}^{(i-1)}]^{-1/2}, \tag{8}$$

则卷积表达式经过化简，可得

$$\boldsymbol{X}^{(i)}=(\boldsymbol{I}-\gamma\times\boldsymbol{L}^{(i-1)})\times\boldsymbol{X}^{(i-1)}. \tag{9}$$

通过以上过程可以扩充用户表示序列 *series*，直至 *series* 的长度达到设定长度 *K*. 由于每一轮的邻接矩阵是变化的，从而每一轮的超图也是在动态变化的. 算法 1 描述了序列生成的全部过程.

**算法 1.** 用户表征序列的动态生成.

输入：原始交互数据 *Dataset*，用户集合 $Set_{\mathrm{user}}$，项目表征矩阵 $\boldsymbol{E}$，原始交互矩阵 $\boldsymbol{H}^{\mathrm{ori}}$，预定长度 *K*，阈值 *gate*；

输出：用户表征序列 *X_list*.

① $\boldsymbol{X}^{(0)}=\boldsymbol{H}^{\mathrm{ori}}\times\boldsymbol{E}$, *X_list* = $\{\boldsymbol{X}^{(0)}\}$; /*初始化*/

② for（*i* from 1 to *K*−1）do /*迭代扩充序列*/

③ *Labels* = *Clustering*($\boldsymbol{X}^{(i-1)}$)；/*对用户聚类获取用户标签*/

④ $Dataset_{\mathrm{extend}}$ = [ ];

⑤ *Clusterset*=*DevideUserByLabel*($Set_{\mathrm{user}}$, *Labels*)；/*以词典形式根据标签生成用户划分*/

⑥ for *cluster* in *Clusterset.keys* do /*在聚类内部进行用户交互扩展*/

⑦ *PairwiseExtendInteractions*(*Clusterset*[cluster], $\boldsymbol{X}^{(i-1)}$, *Dataset*, $Dataset_{\mathrm{extend}}$, *gate*)；

⑧ end for

⑨ $\boldsymbol{H}^{(i-1)}$=*Form_H_from_Data*($Dataset_{\mathrm{extend}}$)；

⑩ Calculate $\boldsymbol{D}_{\mathrm{V}}^{(i-1)}$, $\boldsymbol{D}_{\mathrm{E}}^{(i-1)}$ by $\boldsymbol{H}^{(i-1)}$；

⑪ Calculate$L^{(i-1)}$ by $\boldsymbol{H}^{(i-1)}$, $\boldsymbol{D}_{\mathrm{V}}^{(i-1)}$, $\boldsymbol{D}_{\mathrm{E}}^{(i-1)}$；

⑫ $\boldsymbol{X}^{(i)}$=($\boldsymbol{I}-\gamma\times\boldsymbol{L}^{(i-1)}$)×$\boldsymbol{X}^{(i-1)}$；

⑬ *X_list.append*($\boldsymbol{X}^{(i)}$)；

⑭ end for

⑮ return X_list. /*返回序列*/

### 2.3 注意力机制下的用户表征输出

得到用户表征序列后的工作是将用户表征序列进行加权求和并过滤，最终得到输出的表征. 这个过程可以用式(10)表述为

$$\boldsymbol{X}_{\mathrm{fin}}=\sum_{i=0}^{K-1}\boldsymbol{W}_i\times\boldsymbol{X}^{(i)},$$
$$\boldsymbol{X}_{\mathrm{feature}}=MLP_2(\boldsymbol{X}_{\mathrm{fin}}), \tag{10}$$

其中 $\boldsymbol{W}_i$ 表示序列中的 $\boldsymbol{X}^{(i)}$ 的对应权重，$MLP_2$ 表示模型中使用的第 2 多层感知机，是一个由多组线性层堆叠生成的网络层. 表示序列对应的权重 $\boldsymbol{W}_i$ 是根据注意力机制决定的.

JK 注意力机制[35]的目的是在更大尺度上为每个节点构建扩展特征，从而对序列值对应的权重学习起作用. 模型构建的均值差跳跃知识（average difference jump knowledge, AD-JK）注意力机制希望在构建更广特征域的同时，提升各节点注意力信息的差异性，使序列中每个值感知其与序列均值的矢量距离. 具体做法是将每一个值与整个序列均值的差值作为该值的额外信息输入至第 1 多层感知机，并作为用户表征序列中每个值的特征扩展. 从用户维度来看，扩展后的用户表示输入至权重激活层，并得到权重向量的输出. 在获得了全部权重向量的基础上，对所有权重向量进行 Softmax 处理，获得最终的权重向量序列.

$$\tilde{\boldsymbol{X}}_i^{(l)}=\boldsymbol{X}_i^{(l)}\|\tilde{\boldsymbol{E}}_i^{(l)}, \tag{11}$$

$$\tilde{\boldsymbol{w}}_i(l)=\delta(\tilde{\boldsymbol{X}}_i^{(l)}\times\boldsymbol{s}), \tag{12}$$

$$\boldsymbol{w}_i(l)=\mathrm{e}^{\tilde{\boldsymbol{w}}_i(l)}/\sum_{k=0}^{K-1}\mathrm{e}^{\tilde{\boldsymbol{w}}_i(k)}, \tag{13}$$

其中 $\tilde{\boldsymbol{E}}_i^{(l)}=MLP_1\left(\frac{1}{K}\sum_{k=0}^{K-1}\boldsymbol{X}_i^{(k)}-\boldsymbol{X}_i^{(l)}\right)$ 为第 1 多层感知机的输出结果，$\delta(\cdot)$是激活函数，$\boldsymbol{s}$ 是可学习的向量. 将序列值与均值的矢量距离加入注意力信息简化了第 1 多层感知机的输入规模，精简了结构，使其更加轻量.

通过注意力机制可以从用户的维度输出权重向量；在表示矩阵上进行操作时，可以得到表示矩阵对应的权重矩阵；通过加权和的过滤，得到最终的向量表征矩阵.

### 2.4 候选项评分与排序

在完成编码步骤后，可以根据用户的表示与候选项的表示来计算用户对候选项的评分. 本文所述方法采用内积操作，以 2 个向量的内积结果作为评分输出. 这要求最终输出的用户表示和项目表示具有相同的维度，而本文方法将用户和项目映射至相同的表征空间，很好地满足了这个要求. 对任意用户 *u* 和项目 *d*，且 $u\in Set_{\mathrm{user}}$, $d\in Set_{\mathrm{item}}$, *u* 对 *d* 的评分可以计算为

$$\boldsymbol{s}_{u,d}=\boldsymbol{X}_{\mathrm{feature},u}\cdot(\boldsymbol{E}_d)^{\mathrm{T}}, \tag{14}$$

其中 $\boldsymbol{X}_{\mathrm{feature},u}$ 表示从用户表征矩阵中抽取 *u* 的表示向量，$\boldsymbol{E}_d$ 表示从项目表征矩阵中抽取 *d* 的表示向量. 则对任意用户 *u* 和候选项集 *D*，且满足相关条件 $u\in Set_{\mathrm{user}}$, $D\subseteq Set_{\mathrm{item}}$, *u* 对 *D* 的评分向量可以计算为

$$\boldsymbol{s}_{u,D}=\boldsymbol{X}_{\mathrm{feature},u}\times(\boldsymbol{E}_D)^{\mathrm{T}}, \tag{15}$$

其中 $\boldsymbol{E}_D$ 表示根据候选项集中项目抽取的候选项表

示矩阵.

### 2.5 模型训练与参数优化

在训练阶段，模型在训练集上进行多个轮次的随机采样生成训练集，在每个轮次上计算贝叶斯个性化排名[9]损失(BPR loss)并根据损失函数优化模型中的参数.

具体地，首先在训练集上进行随机采样. 在采样阶段需要根据用户的交互记录以 [用户，正例，负例] 的方式进行测试用例采集，其中正例为在训练集上与用户发生了交互的项目，负例为在训练集上未与用户发生交互的项目. 结合数据集的规模进行上述方式的采集后，可以在训练集上进行模型损失的计算，即

$$Loss=-\sum_{u,pos,neg\in Sample}\ln\sigma(\boldsymbol{s}_{u,\mathrm{pos}}-\boldsymbol{s}_{u,\mathrm{neg}})+\lambda\sum\|\boldsymbol{\Theta}\|^2, \tag{16}$$

其中 *Sample* 表示采样训练集，pos 和 neg 表示一个测试用例里的正例和负例，$\boldsymbol{\Theta}$表示模型中所有多层感知机的网络层的参数，$\lambda$ 控制参数的 L2 正则化的强度. 在模型训练阶段采用 Adam 优化器，并以 mini-batch 的方式对模型进行训练.

### 2.6 模型分析

1)可解释性分析. 用户表征的可解释性依赖于项目表征的可解释性. 在构建用户表征序列时，模型根据用户的交互数据进行初始化. 而在进行交互扩展时，用户将“看到”同一聚类中的其他用户的交互，根据扩展得到的交互构建同质超图，并根据卷积结果构建新的聚类. 通过这样的方法，将用户和项目映射到相同的表征空间. 而项目的表征空间可以有多种方法确定可解释性，如项目标签的独热或多热向量，或 word2vec 等其他自然语言处理模型输出的词向量模型. 结合项目表示向量和交互记录，用户表征向量可以被理解为表示了“某个内容领域有多大兴趣”的用户，可解释性得到了加强.

目前主要的图与超图模型多采用拓扑关系学习用户和项目的向量表示，将用户和项目表征作为模型参数，将维度作为模型的超参数，以此在训练中进行表征的优化，使表征分布符合用户的拓扑特征. 在高度稀疏数据场景下用户不具备鲜明拓扑特征时，模型往往达不到满意的效果.

2)轻量化分析. 本文提出的模型中主要的网络层来自注意力机制和过滤层的第 1 与第 2 多层感知机. 模型的用户表征序列生成部分采用了不使用网络层与激活函数的卷积方法，如式(6)~(9)所示，从而帮助实现用户表征矩阵的快速迭代. 在式(11)~(13)所示的均值差 JK 注意力机制中，第 1 多层感知机的输入维度比原始 JK 注意力机制要小，能够降低模型网络层的复杂度与整体的训练成本. 同时，对比将用户和项目表征作为模型参数的方法，在面对大规模数据集时，模型需要更新优化的参数规模要小得多. 有关轻量化分析将在实验部分通过模型训练效率和网络参数规模对比进行验证.

## 3 实验结果与分析

本节将对实验阶段使用的数据集以及实验结果部分进行分析讨论. 在说明使用数据集时，将详细讨论数据集的特点以及其给模型带来的挑战；实验结果与分析的讨论部分将介绍实验评价指标、基准模型、实验环境、超参设置以及实验结果的分析等 5 个方面的内容.

### 3.1 实验数据

本文使用的数据集是来自腾讯微信的非公开数据集(Wechat-Set). 微信是国内使用最广泛的社交平台应用之一，整合了公众日常生活中应用的众多入口，每天产生海量的用户数据. 使用的微信数据集主要是针对用户在生态内容搜索引擎“搜一搜”内发出的搜索请求和点击阅读的文档的记录，通过数据搜集生成了时间跨度大约 1 个月的查询记录，涉及 101 852 个匿名处理的用户和 27 982 299 个请求关键字与点击阅读的文档标题，并经过清洗形成这个高度稀疏的数据集. 表 1 展示了选取的非公开数据集和常用公开数据集的参数对比. 其中的 Wechat-Set Sample 数据集是从 Wechat-Set 交互记录中随机采样得到的子集，涉及 2 026 个用户在 1 212 934 个文档上产生的 1 672 129

**Table 1 Details and Comparison of Parameters of Experimental Datasets**

**表 1 实验数据集参数详情与对比**

| 参数 | Wechat-Set | Wechat-Set Sample | Movielens-100K | CiteUlike-A | Gowalla | Yelp2018 |
|---|---|---|---|---|---|---|
| 用户数 | 101 852 | 2 026 | 943 | 3 277 | 29 858 | 31 668 |
| 项目数 | 27 982 299 | 1 212 934 | 1 682 | 16 807 | 40 981 | 38 048 |
| 交互记录数 | 11 557 294 | 1 672 129 | 100 000 | 178 062 | 1 027 370 | 1 561 406 |

个交互记录. 数据集中，将用户阅读了某文档作为一次交互.

从表 1 可知，相较于经典的公开数据集，腾讯数据集具有非常高的交互稀疏度（交互记录/(项目×用户)）和比密度（项目/用户）. 这反映了微信生态内数据的高度稀疏以及项目数远大于用户数的特点，也对生态内的推荐算法提出了挑战，即如何完成高度稀疏数据场景下的推荐任务并保证输出的效果和学习得到的用户表示的可解释性. 本文方法在这方面的性能与效果将在实验阶段进行更加详细的说明.

### 3.2 评价指标与基准模型

实验主要在测试集上计算 3 个基础的评价指标来确定本文模型和基准模型的效果对比. 采用的评价指标为精准度（*Precision@TOPK*）、召回率（*Recall@TOPK*）和 *F*1 分数（*F*1-*Score*）.

精准度和召回率在推荐领域属于较为重要的基础概念. 精准度指检索出的相关项目数与检索出的项目总数的比值，衡量的是方法的查准率；而召回率指检索出的相关项目数与相关项目总数的比值，衡量的是方法的查全率. 具体地，精准度和召回率计算为

$$\begin{aligned} Precision@TOPK &= \frac{HIT_IN_TOPK}{TOPK}, \\ Recall@TOPK &= \frac{HIT_IN_TOPK}{Related_Count}, \end{aligned} \tag{17}$$

而 *F* 值为精准度和召回率的调和平均值，较为常用的 *F*1 值的计算方法为

$$F1 = \frac{2 \times Precision@TOPK \times Recall@TOPK}{Precision@TOPK + Recall@TOPK}, \tag{18}$$

从而可以在一定程度上客观地衡量模型的效果.

在基准模型的选取上，本文在实验阶段主要选取了 MF，NGCF，LightGCN 这 3 个方法，通过模型最终的评价指标计算结果来比较模型在选定的数据集上的表现.

1）MF[27]（矩阵分解）. 矩阵分解是较早时期主流的推荐算法，将隐向量的概念加入协同过滤算法中的共现矩阵，从而使模型处理稀疏矩阵的能力得到加强. 该算法期望为每一个用户和项目构成一个隐向量，将用户和项目定位到相同的表征空间上使距离相近的用户具有相似的兴趣特点.

2）NGCF[18]. NGCF 模型研究认为早期矩阵分解等方法中隐藏在用户和项目间的信息难以在编码的过程中表示，编码可能难以捕获协同过滤效果. NGCF 可以在图结构中学习表示，显示地将协同过滤信号放入编码过程并表达高维特征. 但是，模型的节点更新较复杂，且容易在小规模数据上出现过拟合问题.

3）LightGCN[19]. LightGCN 简化了图卷积层的设计，只包含了用于协同过滤的邻域聚合组件. 在用户和项目的交互图上，模型通过线性传播用户和项目的表示来进行表示学习，并根据所有聚积层上得到的嵌入表示输出最终的嵌入生成，模型更简洁且容易实现和训练.

### 3.3 实验环境与基本参数

基准模型与本文模型均在相同的实验环境下进行实验.

基本的硬件与软件参数如表 2 所示. 在基准模型与本文模型的相关参数方面，学习率统一为 0.001，正则化系数为 0.000 1，卷积层数、表征序列长度均为 5；在内存允许的要求下，表示向量维度为 100，所有中间层维度为 128，Light-HGNN 中的多层感知机层数为 2；Light-HGNN 的 $\gamma = 0.1$，聚类数默认为 12，扩展阈值将默认为 0.9.

**Table 2 Experimental Parameters of Software and Hardware**

**表 2 实验软硬件参数**

| 软/硬件 | 具体型号/参数 |
|---|---|
| CPU | Intel® Xeon® Gold 5218 CPU @ 2.30 GHz |
| GPU | GeForce RTX 3090 |
| 内存/GB | 128 |
| 显存/GB | 24 |
| Pytorch 版本 | 1.10.1 |

在测试中，每个用户取头部 20 个项目来进行相关指标计算，所有模型的训练轮次设定为 300，在训练过程中每 10 个轮次进行 1 次模型测试获取指标数据点，生成相关指标的曲线.

### 3.4 实验结果与分析

1）模型结果分析. 模型的评价指标计算结果如表 3 所示. 其中 Light-HGNN（JK）表示采用了原始 JK 注意力机制的模型，Light-HGNN（AD-JK）表示采用了均值差 JK 注意力机制的模型. 对表 3 中的结果分析可知，优化了输入规模的更加精简的均值差 JK 注意力机制与原 JK 注意力机制得到了更好的模型效果，而本文模型的效果，不论在原始 JK 注意力机制还是在改进的均值差 JK 注意力机制下，效果都要好于基准模型.

造成本文模型与基准模型的效果差异的原因可能是数据的高度稀疏特征. 如前所述，腾讯微信的数

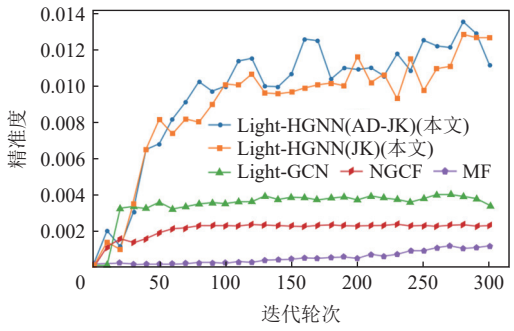
**Table 3 Calculation and Comparison of Evaluation Indicators of Models**

**表 3 模型评价指标计算与对比** ‰

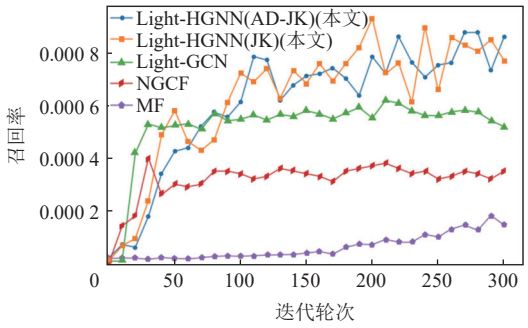
| 模型 | *Precision*@20 | *Recall*@20 | *F*1 值 |
|---|---|---|---|
| MF | 1.234 0 | 0.144 2 | 0.258 2 |
| NGCF | 3.953 4 | 0.479 6 | 0.855 4 |
| LightGCN | 4.047 4 | 0.582 1 | 1.017 8 |
| Light-HGNN（JK）（本文） | 12.660 | 0.851 5 | 1.595 7 |
| Light-HGNN（AD-JK）（本文） | 13.530 | 0.878 6 | 1.650 1 |

据集拥有相对更低的交互密度和相对更高的比密度. 这导致数据集中的很多项目的交互用户很少甚至是唯一的，与此同时，对用户群体而言，同时存在多个用户与 1 个项目交互的情况也是极少的. 这使得依靠用户与项目的拓扑关系很难挖掘出独特的结构特征，且很多项目节点的表征很难在迭代中进行更新，此时将用户表征和项目表征同时作为参数进行优化时会带来较大的不确定性. 本文模型结合了超图与源数据中的拓扑关系并以内容为导向、以项目表征来学习用户的表示，能够使用户的表示更好地反映用户的兴趣分布，从而提高模型的效果.

图 6 和图 7 展示了各模型在实验数据集上取得的相关指标的变化曲线. Light-HGNN 在各项指标上要优于其他基准模型，在精准度指标上的优势更加




(a) 精准度变化情况比较




(b) 召回率变化情况比较

Fig. 6 Variation curve of precision and recall

图 6 精准度和召回率变化曲线

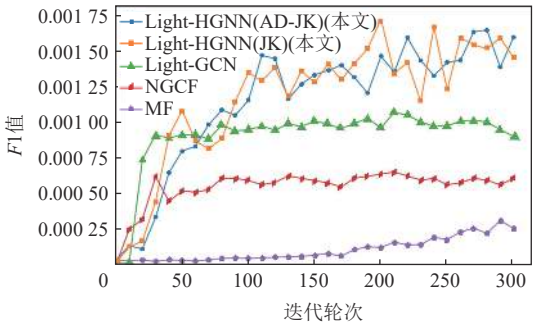



Fig. 7 Variation curve of *F*1 value

图 7 *F*1 值变化曲线

明显. 除前文所述的相关分析，在模型的结构设计上，根据用户聚类进行交互扩展并卷积的做法能使用户直接“感受”到更多的感兴趣的项目；相较于 Light-GCN 对表征序列直接求平均的方法，加入的注意力机制能够更好地挖掘隐藏在用户表征序列中的用户特征，并对输出进行适应性调整，从而提高模型的效果.

2)模型轻量化分析. 表 4 展示了相同软/硬件条件下的模型训练时间与参数规模对比来对模型进行轻量化分析. 从表 4 中可以看出，Light-HGNN 的轮次训练平均时长要远小于 Light-GCN 和 NGCF. 这 2 个模型将用户和项目的表征作为参数进行优化，而 Light-HGNN 的模型设计主要优化模型的第 1、第 2 多层感知机和权重激活层的参数. 因此，根据数据集内的项目数和用户数，Light-GCN 和 NGCF 将产生海量需要更新的参数，参数规模远大于 Light-HGNN 的参数规模. 每轮训练时间的比较也证明了本文模型的轻量化特点.

**Table 4 Comparison of Training Time and Optimized Scale of Parameters**

**表 4 模型训练时间与参数优化规模对比**

| 模型 | 每轮训练时间/s | 模型参数优化个数 |
|---|---|---|
| MF | 189.53 | 121 496 000 |
| NGCF | 480.56 | 121 496 000 |
| Light-GCN | 378.43 | 121 496 000 |
| Light-HGNN（JK）（本文） | 187.24 | 73 156 |
| Light-HGNN（AD-JK）（本文） | 185.82 | 45 728 |

## 4 结 论

本文针对微信“搜一搜”的圈层内容推荐任务提出了一个新的轻量同质超图神经网络模型. 模型采用同质超图对用户数据进行建模，简化了模型设计、降低了复杂度、提高了表征的可解释性. 在序列值的

迭代生成过程中，根据提出的均值差JK注意力机制为每个序列值生成权重矩阵，对序列进行加权求和，过滤后得到用户表示，通过用户表示和项目表示的内积得到用户评分，实现了高度稀疏数据条件下的圈层内容推荐任务. 在真实数据集上进行的实验验证了模型更优的效果. 在后续研究中，将继续探索更有效的注意力机制来进一步提升模型的效果，并对存在时序属性的交互数据的建模进行研究.

**作者贡献声明**：李挺撰写论文，设计实验及测试系统；金福生负责理论和研究框架设计指导及论文撰写；李荣华指导论文撰写；王国仁负责理论指导；段焕中和路彦雄负责实验指导.

## 参 考 文 献

[1] Zhou Yujia, Dou Zhicheng, Wei Bingzheng, et al. Group based personalized search by integrating search behaviour and friend network[C] //Proc of the 44th Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2021: 92−101

[2] Yu Junliang, Yin Hongzhi, Li Jundong, et al. Self-supervised multi-channel hypergraph convolutional network for social recommendation[C] //Proc of the Web Conf 2021. New York: ACM, 2021: 413−424

[3] Wang Wen, Zhang Wei, Rao Jun, et al. Group-aware long- and short-term graph representation learning for sequential group recommendation[C] //Proc of the 43rd Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2020: 1449−1458

[4] Gao Chen, Wang Xiang, He Xiangnan, et al. Graph neural networks for recommender system[C] //Proc of the 15th ACM Int Conf on Web Search and Data Mining. New York: ACM, 2022: 1623−1625

[5] Ma Shuai, Liu Jianwei, Zuo Xin. Survey on graph neural network[J]. Journal of Computer Research and Development, 2022, 59(1): 47−80 (in Chinese)
(马帅，刘建伟，左信. 图神经网络综述[J]. 计算机研究与发展，2022, 59(1): 47−80)

[6] Xia Xin, Yin Hongzhi, Yu Junliang, et al. Self-supervised hypergraph convolutional networks for session-based recommendation[C] //Proc of the 45th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2021: 4503−4511

[7] Wu Jiancan, Wang Xiang, Feng Fuli, et al. Self-supervised graph learning for recommendation[C] //Proc of the 44th Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2021: 726−735

[8] Ji Shuyi, Feng Yifan, Ji Rongrong, et al. Dual channel hypergraph collaborative filtering[C] //Proc of the 26th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2020: 2020−2029

[9] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback[J]. arXiv preprint, arXiv: 1205. 2618, 2012

[10] Feng Yifan, You Haoxuan, Zhang Zizhao, et al. Hypergraph neural networks[C] //Proc of the 43rd AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2019: 3558−3565

[11] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint, arXiv: 1609. 02907, 2016

[12] Zhang Ruochi, Zou Yuesong, Ma Jian. Hyper-SAGNN: A self-attention based graph neural network for hypergraphs[J]. arXiv preprint, arXiv: 1911. 02613, 2019

[13] Bai Song, Zhang Feihu, Torr P H S. Hypergraph convolution and hypergraph attention[J]. arXiv preprint, arXiv: 1901. 08150, 2020

[14] Hu Youpeng, Li Xunkai, Wang Yujie, et al. Adaptive hypergraph auto-encoder for relational data clustering[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(3): 2231−2242

[15] Teevan J, Liebling D J, Ravichandran G. Understanding and predicting personal navigation[C] //Proc of the 4th ACM Int Conf on Web Search and Data Mining. New York: ACM, 2011: 85−94

[16] Dou Zhicheng, Song Ruihua, Wen Jirong. A large-scale evaluation and analysis of personalized search strategies[C]//Proc of the 16th Int Conf on World Wide Web. New York: ACM, 2007: 581−590

[17] Carman M J, Crestani F, Harvey M, et al. Towards query log based personalization using topic models[C] //Proc of the 19th ACM Int Conf on Information and Knowledge Management. New York: ACM, 2010: 1849−1852

[18] Wang Xiang, He Xiangnan, Wang Meng, et al. Neural graph collaborative filtering[C]//Proc of the 42nd Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2019: 165−174

[19] He Xiangnan, Deng Kuan, Wang Xiang, et al. LightGCN: Simplifying and powering graph convolution network for recommendation [C] //Proc of the 43rd Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2020: 639−648

[20] Ai Qingyao, Zhang Yongfeng, Bi Keping, et al. Explainable product search with a dynamic relation embedding model[J]. ACM Transactions on Information Systems, 2019, 38(1): 1−29

[21] Liu Shang, Gu Wanli, Cong Gao, et al. Structural relationship representation learning with graph embedding for personalized product search[C] //Proc of the 29th ACM Int Conf on Information & Knowledge Management. New York: ACM, 2020: 915−924

[22] Hu Binbin, Shi Chuan, Zhao W X, et al. Leveraging meta-path based context for top-*n* recommendation with a neural co-attention model[C] //Proc of the 24th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2018: 1531−1540

[23] Ungar L H, Foster D P. Clustering methods for collaborative filtering[C]//Proc of the 15th AAAI Workshop on Recommendation Systems. Palo Alto, CA: AAAI, 1998: 114−129

[24] Yang Xiwang, Guo Yang, Liu Yong. Bayesian-inference-based recommendation in online social networks[J]. IEEE Transactions on Parallel and Distributed Systems, 2012, 24(4): 642−651

[25] Lee C H, Kim Y H, Rhee P K. Web personalization expert with combining collaborative filtering and association rule mining technique[J]. Expert Systems with Applications, 2001, 21(3): 131−137

[26] Paterek A. Improving regularized singular value decomposition for collaborative filtering[C]//Proc of the 13th KDD Cup and Workshop. New York: ACM, 2007: 5−8

[27] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30−37

[28] He Xiangnan, Liao Lizi, Zhang Hanwang, et al. Neural collaborative filtering[C]//Proc of the 26th Int Conf on World Wide Web. Geneva, Switzerland: IW3C2, 2017: 173−182

[29] Jeh G, Widom J. SimRank: A measure of structural-context similarity[C]//Proc of the 8th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2002: 538−543

[30] Gori M, Pucci A, Roma V, et al. ItemRank: A random-walk based scoring algorithm for recommender engines[C] //Proc of the 20th Int Joint Conf on Artificial Intelligence. San Francisco, CA: Margan Kaufmann, 2007: 2766−2771

[31] He Xiangnan, Gao Ming, Kan M Y, et al. BiRank: Towards ranking on bipartite graphs[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 29(1): 57−71

[32] Berg R, Kipf T N, Welling M. Graph convolutional matrix completion[J]. arXiv preprint, arXiv: 1706. 02263, 2017

[33] Ying R, He Ruining, Chen Kaifeng, et al. Graph convolutional neural networks for web-scale recommender systems[C] //Proc of the 24th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2018: 974−983

[34] Hamilton W, Ying R, Leskovec J. Inductive representation learning on large graphs[C] //Proc of the 31st Int Conf on Neural Information Processing Systems. New York: Curran Associates, 2017: 1025−1035

[35] Zhang Wentao, Yin Ziqi, Sheng Zeang, et al. Graph attention multi-layer perceptron[J]. arXiv preprint, arXiv: 2108. 10097, 2021



**Li Ting**, born in 1996. Master candidate. Student member of CCF. His main research interest includes graph neural network.

**李　挺**, 1996 年生. 硕士研究生. CCF 学生会员. 主要研究方向为图神经网络.



**Jin Fusheng**, born in 1977. PhD, associate professor, master supervisor. His main research interests include software architecture, machine learning, and deep learning.

**金福生**, 1977 年生. 博士, 副教授, 硕士生导师. 主要研究方向为软件架构、机器学习、深度学习.



**Li Ronghua**, born in 1985. PhD, professor, PhD supervisor. His main research interests include graph data management and mining, graph computing system, and graph neural network.

**李荣华**, 1985 年生. 博士, 教授, 博士生导师. 主要研究方向为图数据管理与挖掘、图计算系统、图神经网络.



**Wang Guoren**, born in 1966. PhD, professor, PhD supervisor. His main research interests include uncertain data management and data intensive computing.

**王国仁**, 1966 年生. 博士, 教授, 博士生导师. 主要研究方向为不确定数据管理、数据密集型计算.



**Duan Huanzhong**, born in 1985. Master. His main research interests include machine learning and data mining.

**段焕中**, 1985 年生. 硕士. 主要研究方向机器学习、数据挖掘.



**Lu Yanxiong**, born in 1984. Master. His main research interests include natural language processing and machine learning.

**路彦雄**, 1984 年生. 硕士. 主要研究方向为自然语言处理、机器学习.