

## 融合层注意力机制的多视角图对比学习推荐方法

钱忠胜 黄 恒 朱 辉 刘金平

(江西财经大学计算机与人工智能学院 南昌 330013)

([changesme@163.com](mailto:changesme@163.com))

## Multi-Perspective Graph Contrastive Learning Recommendation Method with Layer Attention Mechanism

Qian Zhongsheng, Huang Heng, Zhu Hui, and Liu Jinping

(School of Computer and Artificial Intelligence, Jiangxi University of Finance and Economics, Nanchang 330013)

**Abstract** Graph contrastive learning is widely employed in recommender system due to its effectiveness in mitigating data sparsity issue. However, most current recommendation algorithms based on graph contrastive learning start to learn from only a single perspective, severely limiting the model's generalization capability. Furthermore, the over-smoothing problem inherent in graph convolutional networks also affects the model's stability. Based on this, we propose the multi-perspective graph contrastive learning recommendation method with layer attention mechanism. On the one hand, this method proposes three contrastive learning approaches from two different perspectives. From a view-level perspective, it constructs perturbation-enhanced view by adding random noise for the original graph and employing singular value decomposition (SVD) recombination to establish SVD-enhanced view. It then performs view-level contrastive learning on these two enhanced views. From a node-level perspective, it conducts contrastive learning on candidate nodes and candidate structural neighbors using semantic information between nodes, optimizes multi-task learning with three contrastive auxiliary tasks and a recommendation task to enhance the quality of node embeddings, thereby improving the model's generalization ability. On the other hand, in the context of learning for user and item node embeddings by graph convolutional network, a layer attention mechanism is employed to aggregate the final node embeddings. This enhances the model's higher-order connectivity and mitigates the over-smoothing issue. When compared with ten classic models on four publicly available datasets, such as LastFM, Gowalla, Ifashion, and Yelp, the results indicate that this method achieves an average improvement of 3.12% in *Recall*, 3.22% in *Precision*, and 4.06% in *NDCG*. This demonstrates the effectiveness of the approach proposed in this work.

**Key words** layer attention mechanism; contrastive learning; graph convolutional network; multi-task learning; recommender system

**摘 要** 图对比学习因其可有效缓解数据稀疏问题被广泛应用在推荐系统中。然而,目前大多数基于图对比学习的推荐算法均采用单一视角进行学习,这极大地限制了模型的泛化能力,且图卷积网络本身存在的过度平滑问题也影响着模型的稳定性。基于此,提出一种融合层注意力机制的多视角图对比学习推荐

收稿日期: 2023-10-10; 修回日期: 2024-04-23

基金项目: 国家自然科学基金项目(62262025); 江西省自然科学基金重点项目(20224ACB202012); 赣鄱俊才支持计划-主要学科学术和技术带头人培养项目-领军人才(学术类)(20243BCE51024)

This work was supported by the National Natural Science Foundation of China (62262025), the Key Program of the Natural Science Foundation of Jiangxi Province of China (20224ACB202012), and Jiangxi Poyang Support Program for Talents of China-Major Discipline Academic and Technical Leaders Training Program of Jiangxi Province-Leading Talent (Academic) (20243BCE51024).

方法.一方面,该方法提出2种不同视角下的3种对比学习,在视图级视角下,通过对原始图添加随机噪声构建扰动增强视图,利用奇异值分解(singular value decomposition)重组构建SVD增强视图,对这2个增强视图进行视图级对比学习;在节点视角下,利用节点间的语义信息分别进行候选节点和候选结构邻居对比学习,并将3种对比学习辅助任务和推荐任务进行多任务学习优化,以提高节点嵌入的质量,从而提升模型的泛化能力.另一方面,在图卷积网络学习用户和项目的节点嵌入时,采用层注意力机制的方式聚合最终的节点嵌入,提高模型的高阶连通性,以缓解过度平滑问题.在4个公开数据集LastFM, Gowalla, Ifashion, Yelp上与10个经典模型进行对比,结果表明该方法在Recall, Precision, NDCG这3个指标上分别平均提升3.12%, 3.22%, 4.06%,这说明所提方法是有效的.

**关键词** 层注意力机制;对比学习;图卷积网络;多任务学习;推荐系统

**中图法分类号** TP399

**DOI:** 10.7544/issn1000-1239.202330804 **CSTR:** 32373.14.issn1000-1239.202330804

推荐系统作为一种高效的信息过滤方式,对缓解用户在各种Web应用中的信息过载问题起到重要作用.推荐系统背后的核心思想是协同过滤(collaborative filtering, CF)<sup>[1-2]</sup>,早期的工作利用矩阵分解<sup>[3-4]</sup>将每个用户(或项目)投影到嵌入向量中.近年来,图神经网络(graph neural network, GNN)因其突出的表示学习能力,被广泛应用于各种研究中<sup>[5-8]</sup>,并实现了较优的推荐性能.然而,基于GNN的方法易遭受交互数据稀疏或数据嘈杂的影响,使其无法学习高质量的节点嵌入.

最近有研究将对比学习(contrastive learning, CL)与GNN相结合的方式<sup>[9-10]</sup>为用户生成更高质量的推荐,称为图对比学习(graph contrastive learning, GCL).GCL的关键是通过视图生成器生成合适的锚点及正负实例,使得在嵌入空间中锚点靠近正实例而远离负实例,从而提高节点嵌入质量.为构建增强视图,Wu等人<sup>[9]</sup>提出SGL模型,通过使用随机丢弃的增强策略生成正实例.但具有图信息随机丢弃的数据增强,例如随机节点/边丢弃和随机游走等破坏了原有的图结构信息,不能很好地保存语义信息,且这种数据增强策略的泛化能力不强,在不同场景下需人工选择合适的增强策略,限制了模型的适应范围.为改进推荐中的图对比学习,Yu等人<sup>[11]</sup>提出SimGCL模型,采用具有随机噪声扰动的嵌入增强方式.Xia等人<sup>[12]</sup>提出HCCF模型以及Lin等人<sup>[13]</sup>提出NCL模型,从结构与语义空间中提取用户和项目的潜在关系.但这些方法均忽略GNN通常带来的过度平滑现象而导致的模型性能无法进一步提升问题.

尽管基于GCL的方法已经做了一些改进,但依旧存在2个问题:1)随机丢弃的图增强方式可能会丢失重要的节点信息,且目前大多图对比学习的模

型均采用单一的对比方式,这限制了模型的泛化能力.2)目前大多数基于GNN的图对比学习推荐方法均或多或少地受到过度平滑问题的影响,这导致节点无法有效区分.

基于以上分析,本文提出一种融合层注意力机制的多视角图对比学习推荐方法(the multi-perspective graph contrastive learning recommendation method with layer attention mechanism, LA-MPGCL).一方面,通过向原始的图嵌入中添加随机均匀的噪声以实现扰动增强视图,同时通过奇异值分解(singular value decomposition, SVD)提取用户-项目的有效信息构建SVD增强视图,以实施增强视图对比学习(见2.3节);并利用用户和候选项目的不同层嵌入间的关系,基于语义相似度来选择对比学习对象的锚点和正负实例进行节点级对比学习(见2.4节).通过2种不同视角(视图级、节点级)的3种对比学习(增强视图对比、候选节点对比、候选结构邻居对比)来增强模型的泛化能力.另一方面,目前基于GNN的推荐大多采用平均池化的方式聚合最终的节点嵌入,其平等地对待每一层的嵌入表示,但现有研究表明,高阶的邻域信息可能存在负面信息,多层的嵌入传播会使兴趣不同的用户具有相似的嵌入,导致过度平滑现象<sup>[14]</sup>.为此,在生成最终的节点嵌入时,LA-MPGCL模型采用层注意力机制(layer attention mechanism, LA)方式,对每一层的嵌入分配权重,这能有效地缓解过度平滑问题,使模型能充分地利用高阶连通性进行推荐(见2.2节).本文主要工作有3个方面:

1)构建多视角图对比学习,提高模型的泛化能力.将具有随机噪声增强的扰动增强视图与通过SVD的SVD增强视图进行视图级对比学习,同时将基于语义相似度的用户与候选项目实行节点级对比学习.

将2种视角相结合,可学习更高质量的节点嵌入,从而改善推荐效果.

2)融合层注意力机制,有效缓解过度平滑问题. LA-MPGCL模型在生成最终的节点嵌入时,采用层注意力机制关注每一层的权重,从而稳定深层训练,使模型能更好地利用高阶邻域的协同信息,进而缓解过度平滑问题.

3)在4个公开数据集上进行充分的对比实验和消融实验.与当前经典的、主流的图卷积推荐模型以及图对比学习推荐模型相比,LA-MPGCL模型在性能上均占优,同时也验证了模型中各组件的必要性.

## 1 相关工作

近年来,GNN在推荐领域取得了巨大成功<sup>[15-16]</sup>,特别地,图卷积网络(graph convolutional network, GCN)<sup>[17]</sup>成为了GNN中最流行的变体,进一步推动了图神经网络推荐模型的发展<sup>[18-19]</sup>.Berg等人<sup>[20]</sup>提出GCMC模型,通过自动编码器提取一阶邻居节点的信息.相比之下,Wang等人<sup>[5]</sup>提出NGCF模型,通过在图中传播节点嵌入来提取用户和项目间的高阶协同信号.与NGCF相比,He等人<sup>[6]</sup>提出LightGCN模型,通过去除非线性激活函数和特征变换简化模型并获得更优的推荐效果.LightGCN成为目前在推荐系统中最受欢迎的网络结构,基于此发展的模型也越来越多.Ji等人<sup>[21]</sup>提出DHCF模型,采用超图对具有混合高阶相关性的用户和项目进行建模.Liu等人<sup>[14]</sup>提出IMP-GCN模型,将具有相似兴趣的节点进行聚类来进一步增强LightGCN,以减轻负面信息传播给模型带来的影响.为避免深度GNN的信息瓶颈,Chen等人<sup>[22]</sup>提出GOTNet模型,通过非局部注意力建立节点及其邻居节点间的长距离依赖.Huang等人<sup>[23]</sup>提出MixGCF模型,通过Mixup技术将正样本信息注入到随机抽取的负样本中,合成硬负样本(hard negative sample),有效地提高了训练样本的质量.Mao等人<sup>[24]</sup>提出UltraGCN模型,使用约束来估计无限卷积运算后节点的嵌入.Shen等人<sup>[25]</sup>提出GFCF模型,证明了基于邻域的方法、线性自动编码器和低秩矩阵分解是各种经典低通滤波器的特例.

然而,基于GNN的模型通常遭受着数据稀疏的困扰,难以学习到高质量的嵌入表示,限制了模型的泛化能力.尽管融合知识图谱<sup>[26]</sup>、社交关系<sup>[27]</sup>、多行为<sup>[28]</sup>等信息的推荐模型可缓解此问题,但在实际操作中往往难以获得这样的辅助信息.幸运的是,最近

的研究发现CL<sup>[11]</sup>可有效缓解数据的稀疏性而增强模型的泛化能力,进而改善推荐效果.

CL先在计算机视觉等领域引起了广泛关注<sup>[29]</sup>.CL通过最大化正样本对的相似性,有效地提高了嵌入质量.作为一种典型的自监督学习方法,CL不需要人工标记即可挖掘数据的内在特征.结合GNN和CL的优点,基于GCL的推荐算法应运而生.Wu等人<sup>[9]</sup>提出SGL模型,使用随机节点/边丢弃和随机游走在图结构与嵌入上实行数据增强,构建出多个子视图进行对比学习,然而这种随机增强会破坏原有图结构,可能会丢失重要的图结构信息.与SGL不同,Yu等人<sup>[11]</sup>提出SimGCL模型,放弃随机丢弃的图增强方式,而是通过将噪声均匀添加到每一层嵌入以生成正实例,从而创建对比视图.Xia等人<sup>[12]</sup>提出HCCF模型,通过超图增强的跨视图对比学习捕获局部和全局协同关系,从而全面捕捉用户间的高阶依赖关系.Lin等人<sup>[13]</sup>提出NCL模型,采用嵌入空间中相似的语义原型和结构邻居作为正实例.Cai等人<sup>[30]</sup>提出LightGCL模型,使用矩阵分解来重构用户-项目交互图以生成对比实例,可有效利用原图信息.Xia等人<sup>[31]</sup>提出AutoGCL模型,为每种数据增强方法设计了视图生成器,在节点级自适应地选择节点丢弃和属性掩蔽,提升了数据增强的适应性问题.Li等人<sup>[32]</sup>提出GFormer模型,利用图形自动编码器来重建被屏蔽的用户-项目交互以进行数据增强,通过这种方式生成增强的训练数据,模型可学习用户和项目更有效的表示.

上述已有研究表明,GCL已在推荐中得到了一定发展,但大多数研究人员仍关注于如何构造更有效的增强视图,忽略了用户和候选项目间的关系,且采用的对比学习形式单一,模型泛化能力不强.此外,在聚合最终节点表示时,很少有学者关注图卷积自身的过度平滑问题对学习嵌入表示的影响.基于此,本文提出LA-MPGCL模型构造2个视角下的3种对比学习,先利用扰动增强和SVD增强后的视图进行对比学习以保留全局图结构信息,同时采用节点语义信息构建2个节点级对比学习,在生成最终节点嵌入时采用层注意力机制优化聚合函数学习更高质量的嵌入表示,以提升推荐效果.

## 2 LA-MPGCL 模型

下面先给出本文模型LA-MPGCL的整体框架,如图1所示,主要包含4个模块:

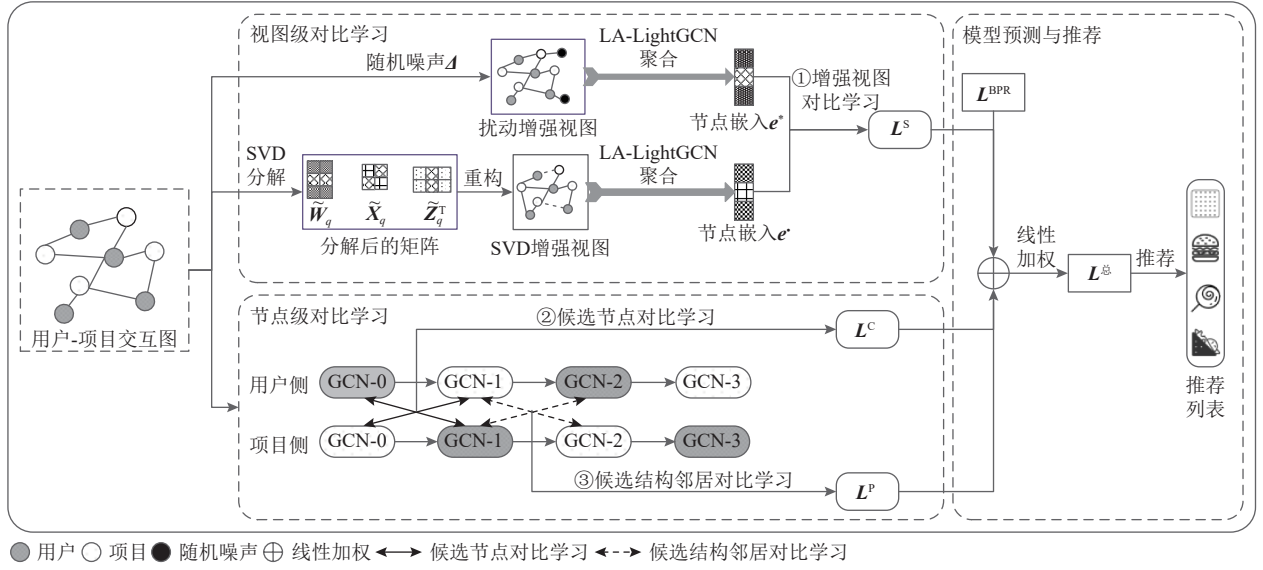


Fig. 1 The whole framework of LA-MPGCL model

图1 LA-MPGCL模型的整体框架

1) 融合层注意力机制的图卷积结构. 利用图卷积网络聚合节点嵌入时, 以层注意力机制代替传统图卷积中的平均池化, 称该网络结构为 LA-LightGCN, 通过这种方式聚合嵌入可考虑不同层间的权重关系, 增强模型的高阶连通性, 进而缓解过度平滑问题.

2) 视图级对比学习. 在视图级对比学习中构建了2种增强视图, 通过添加随机噪声以构建扰动增强视图, 利用SVD重构以构造SVD增强视图, 使用LA-LightGCN聚合最终的节点嵌入表示, 再将2种视图的最终嵌入进行增强视图对比学习, 以充分提取图结构信息.

3) 节点级对比学习. 在节点级对比学习中采用2种对比学习方式, 利用节点交互的语义关系分别建立候选节点对比学习和候选结构邻居对比学习, 以优化节点嵌入表示.

4) 模型预测与推荐. 采用多任务学习的方法训练模型, 推荐任务为主任务, 3种对比学习为辅助任务, 这样来联合优化总损失, 进而提升模型的泛化能力, 达到更优的推荐效果.

## 2.1 问题定义与符号说明

给定用户集  $U = \{u_1, u_2, \dots, u_m\}$ 、项目集  $I = \{i_1, i_2, \dots, i_n\}$  以及它们的交互矩阵  $\mathbf{Y} \in \mathbb{R}^{m \times n}$ , 其中  $m$  和  $n$  分别表示用户数和项目数. 若用户  $u$  与项目  $i$  交互, 则  $y_{ui} = 1$ , 否则  $y_{ui} = 0$ ,  $\mathbf{Y} = (y_{ui})$ . 用户与项目的交互形成图  $G(V, E)$ , 其中节点集  $V = U \cup I$  包含所有的用户和项目节点, 边集  $E = \{y_{ui} | y_{ui} = 1\}$  表示节点的交互. 本文模型的任务是预测用户  $u$  与未交互过的项目  $i$  在未来交互的概率  $\hat{y}_{ui}$ .

为便于阐述, 先对文中一些主要符号进行说明, 如表1所示.

Table 1 Description of Main Symbols

表1 主要符号说明

符号	含义
$U = \{u_1, u_2, \dots, u_m\}$	用户集
$I = \{i_1, i_2, \dots, i_n\}$	项目集
$L$	网络层数
$N_u$	用户 $u$ 的交互项目集
$N_i$	与项目 $i$ 交互的用户集
$e_u^{(0)}, e_i^{(0)}$	用户 $u$ 、项目 $i$ 的初始嵌入
$e_u^{(l)}, e_i^{(l)}$	用户 $u$ 、项目 $i$ 的第 $l$ 层嵌入
$e_u, e_i$	用户 $u$ 、项目 $i$ 的最终嵌入
$\hat{y}_{ui}$	用户 $u$ 与项目 $i$ 交互的概率

## 2.2 融合层注意力机制的图卷积结构

传统图卷积的推荐方法基于用户-项目二部图, 通过聚合邻居节点的表示来更新自身节点的嵌入表示, 而 LightGCN<sup>[6]</sup> 在传播过程中舍弃了传统的特征变化和非线性激活函数, 使其更加轻量并达到更优效果. 然而, 该模型仍存在过度平滑问题, 即堆叠多个 GCN 层会使节点嵌入难以区分从而影响推荐性能. LA 可学习不同 GCN 层的权重以优化最终节点嵌入, 并提升深层学习的稳定性. 为此, 我们将 LightGCN 最终嵌入的聚合函数用 LA 替换, 以稳定在深层 GCN 上的训练, 并使模型能更好地利用高阶连通性进行推荐, 该方法利用融合层注意力机制的轻量级图卷积结构, 称为 LA-LightGCN. 下面以聚合用户  $u_1$  的过

程为例,如图2所示。

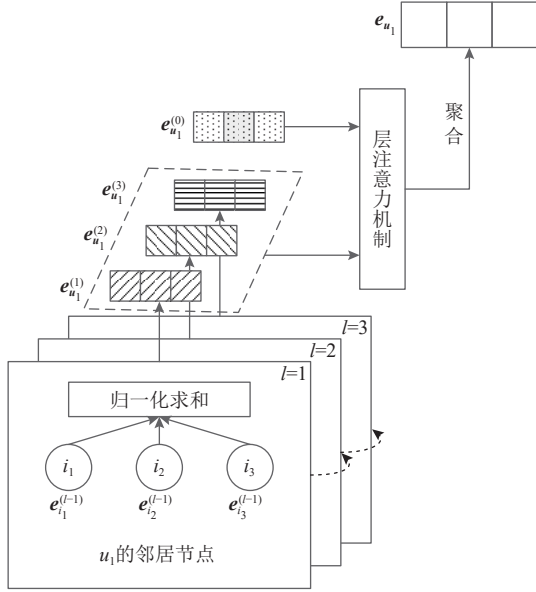


Fig. 2 Graph convolutional structure with layer attention mechanism

图2 融合层注意力机制的图卷积结构

给定用户  $u$  和项目  $i$ , 根据 LightGCN<sup>[6]</sup> 的结构, 我们丢弃传播函数中的非线性激活和特征变换, 节点特征向量的聚合表达式如式(1)所示:

$$\begin{aligned} e_u^{(l)} &= \sum_{i \in N_u} \frac{1}{\sqrt{|N_u||N_i|}} e_i^{(l-1)}, \\ e_i^{(l)} &= \sum_{u \in N_i} \frac{1}{\sqrt{|N_u||N_i|}} e_u^{(l-1)}, \end{aligned} \quad (1)$$

其中  $e_u^{(l)}$  和  $e_i^{(l)}$  分别表示在  $l$  层传播后用户  $u$  和项目  $i$  的嵌入表示.  $N_u$  表示用户  $u$  交互的项目集,  $N_i$  表示与项目  $i$  交互的用户集.  $\frac{1}{\sqrt{|N_u||N_i|}}$  表示对称归一化项, 其作用是避免嵌入的规模随着图卷积的增加而增大.

与 LightGCN 不同的是, 在获得每层嵌入表示后, 最终用户/项目的嵌入表示通过层注意力机制获得, 如式(2)所示:

$$\begin{aligned} e_u &= \sum_{l=0}^L \alpha_u^{(l)} e_u^{(l)}, \\ e_i &= \sum_{l=0}^L \alpha_i^{(l)} e_i^{(l)}, \end{aligned} \quad (2)$$

其中  $e_u$  和  $e_i$  分别表示用户  $u$ 、项目  $i$  的最终嵌入表示,  $L$  是网络层数. 经过  $L$  层传播后, 图中的节点可聚合来自高阶邻域的信息. 这里,  $\alpha_u^{(l)}$  和  $\alpha_i^{(l)}$  分别是聚合用户嵌入与项目嵌入时第  $l$  层的注意力权重, 如式(3)所示:

$$\begin{aligned} \alpha_u^{(l)} &= \frac{\exp(\langle W_{\text{Att}}, e_u^{(l)} \rangle)}{\sum_{l=0}^L \exp(\langle W_{\text{Att}}, e_u^{(l)} \rangle)}, \\ \alpha_i^{(l)} &= \frac{\exp(\langle W_{\text{Att}}, e_i^{(l)} \rangle)}{\sum_{l=0}^L \exp(\langle W_{\text{Att}}, e_i^{(l)} \rangle)}, \end{aligned} \quad (3)$$

其中  $W_{\text{Att}} \in \mathbb{R}^d$  是计算注意力权重的参数,  $d$  为嵌入向量维度.

### 2.3 视图级对比学习

为最大程度地保留并充分利用原有的图结构信息, 我们通过添加随机噪声构建扰动增强视图, 利用 SVD 提取有效信息构建 SVD 增强视图, 进行增强视图对比学习.

#### 2.3.1 扰动增强视图构建

基于深度学习的推荐系统通过学习更高质量的嵌入表示来提升推荐效果, 现有研究表明随机丢弃节点/边的视图增强方式很可能会破坏原有的重要图结构信息<sup>[11]</sup>. 文献 [33] 通过理论分析和实验证明了嵌入之间的对齐性和分布的均匀性均可能影响模型的性能. 其中对齐性表示在高维空间中相关的嵌入彼此接近, 而均匀性表示在整个空间中不相同的嵌入表征分布应均匀.

由上可知, 通过调节学习到的表征分布的均匀性可使模型达到更优性能. 为实现这一目标, 我们在原始表征中随机添加均匀噪声扰动, 实现表征级数据增强. 这种增强方式可使嵌入表征在整个空间分布的均匀性更可控, 既避免了语义偏差, 又保存了原始表征中可学习的信息. 其添加过程是在第  $l$  层聚合后执行式(1)时增加扰动, 如式(4)所示:

$$\begin{aligned} e_u^{(l)*} &= e_u^{(l)} + \Delta_u^{(l)*}, \\ e_i^{(l)*} &= e_i^{(l)} + \Delta_i^{(l)*}, \end{aligned} \quad (4)$$

其中  $e_u^{(l)*}$  和  $e_i^{(l)*}$  分别是加入了扰动后的第  $l$  层用户嵌入与项目嵌入,  $\Delta_u^{(l)*}$  和  $\Delta_i^{(l)*}$  分别是加入到第  $l$  层用户嵌入与项目嵌入的扰动, 其约束条件为  $\|\Delta\|_2 = \varepsilon$  与  $\Delta = \bar{\Delta} \odot \text{sgn}(e^{(l)})$ ,  $\bar{\Delta} \in \mathbb{R}^d \sim U(0, 1)$ .  $\|\Delta\|_2 = \varepsilon$  控制了噪声的大小,  $\varepsilon$  是一个超参数, 在不同的数据集上会有不同的表现, 关于其取值将在实验部分讨论.  $\Delta = \bar{\Delta} \odot \text{sgn}(e^{(l)})$  中的  $\odot$  表示同或运算,  $\text{sgn}$  是符号函数, 保证加入的噪声维度与原嵌入维度一致, 这样加入的噪声就不会与原嵌入有较大偏差.

经过这一扰动添加操作, 当前的节点嵌入就增加了随机均匀的噪声扰动.

#### 2.3.2 SVD 增强视图构建

通过随机扰动增强的视图已尽可能地保留原有

图结构信息,为充分利用有效信息,我们采取 SVD 提取重要的协同信息.具体地,先对归一化后的邻接矩阵  $\mathbf{A}$  执行 SVD 分解,得到  $\mathbf{A} = \mathbf{W}\mathbf{X}\mathbf{Z}^T$ , 其中  $\mathbf{W}$  和  $\mathbf{Z}$  分别是  $m \times m$  阶和  $n \times n$  阶的正交矩阵,  $\mathbf{X}$  是  $m \times n$  阶的对角矩阵,其对角线上的元素即为  $\mathbf{A}$  的奇异值.通常,保留较大的奇异值、丢弃较小的奇异值进行去噪.基于此,我们截取最大的前  $q$  个奇异值,抛弃剩余较小的,并重构邻接矩阵.  $\tilde{\mathbf{A}} = \mathbf{W}_q \mathbf{X}_q \mathbf{Z}_q^T$ , 其中  $\mathbf{W}_q \in \mathbb{R}^{m \times q}$ ,  $\mathbf{Z}_q \in \mathbb{R}^{n \times q}$  分别表示  $\mathbf{W}$  和  $\mathbf{Z}$  的前  $q$  列.  $\mathbf{X}_q \in \mathbb{R}^{q \times q}$  是  $q$  个最大奇异值的对角矩阵.重构后的邻接矩阵  $\tilde{\mathbf{A}}$  是原邻接矩阵  $\mathbf{A}$  的低秩近似.

经过 SVD 分解的图结构学习有 2 方面的优势.一方面,该方法通过识别用户偏好中更重要的信息作为重构图的主要成分;另一方面,考虑了每对用户和项目的潜在关联,更能挖掘全局信息.这样重构后的图结构可使模型嵌入的表征分布更加合理,可惜的是,在大型矩阵上执行精确的 SVD 需要大量的计算资源,这在处理大规模数据时则显得不切实际.因此,我们采用文献 [34] 提出的随机 SVD 算法,先用低秩的正交矩阵逼近输入矩阵的范围,再对这个较小的矩阵执行 SVD 分解,如式(5)所示:

$$\begin{aligned} \tilde{\mathbf{W}}_q, \tilde{\mathbf{X}}_q, \tilde{\mathbf{Z}}_q^T &= \text{ApproxSVD}(\mathbf{A}, q), \\ \tilde{\mathbf{A}}_{\text{SVD}} &= \tilde{\mathbf{W}}_q \tilde{\mathbf{X}}_q \tilde{\mathbf{Z}}_q^T, \end{aligned} \quad (5)$$

其中  $q$  是分解所需的秩,参照文献 [30],  $q = 5$ ,  $\tilde{\mathbf{W}}_q \in \mathbb{R}^{m \times q}$ ,  $\tilde{\mathbf{X}}_q \in \mathbb{R}^{q \times q}$ ,  $\tilde{\mathbf{Z}}_q \in \mathbb{R}^{n \times q}$  分别是  $\mathbf{W}_q, \mathbf{X}_q, \mathbf{Z}_q$  的近似.我们将重构后的矩阵输入传播表达式中,如式(6)所示:

$$\begin{aligned} \mathbf{e}_u^{(l)*} &= \tilde{\mathbf{A}}_{\text{SVD}} \mathbf{e}_i^{(l-1)} = \tilde{\mathbf{W}}_q \tilde{\mathbf{X}}_q \tilde{\mathbf{Z}}_q^T \mathbf{e}_i^{(l-1)}, \\ \mathbf{e}_i^{(l)*} &= \tilde{\mathbf{A}}_{\text{SVD}}^T \mathbf{e}_u^{(l-1)} = \tilde{\mathbf{Z}}_q \tilde{\mathbf{X}}_q \tilde{\mathbf{W}}_q^T \mathbf{e}_u^{(l-1)}, \end{aligned} \quad (6)$$

其中  $\mathbf{e}_u^{(l)*}$  和  $\mathbf{e}_i^{(l)*}$  分别是重构后的图结构中的用户嵌入和项目嵌入.通过这种方式,我们不需要计算和储存大型的矩阵  $\tilde{\mathbf{A}}_{\text{SVD}}$ , 只需储存低维的  $\tilde{\mathbf{W}}_q, \tilde{\mathbf{X}}_q, \tilde{\mathbf{Z}}_q$ , 并在 SVD 分解的预处理阶段提前计算,极大地提升了模型效率.

### 2.3.3 增强视图对比学习

通过对原有图结构添加噪声扰动和 SVD 重构,得到 2 个增强视图,在尽可能保留原有图结构信息的同时,也充分利用了图结构中的有效信息.基于此,构造扰动增强视图与 SVD 增强视图的对比学习,将同一个节点在不同视图下的嵌入作为正例对  $(\mathbf{e}_u^*, \mathbf{e}_u^*)$ , 将不同节点在不同视图下的嵌入作为负例对  $(\mathbf{e}_u^*, \mathbf{e}_v^*)$ . 其用户的损失表示,如式(7)所示:

$$L_u^S = - \sum_{u \in U} \ln \frac{\exp(\text{sim}(\mathbf{e}_u^*, \mathbf{e}_u^*)/\tau)}{\sum_{v \in U, v \neq u} \exp(\text{sim}(\mathbf{e}_u^*, \mathbf{e}_v^*)/\tau)}, \quad (7)$$

其中  $L_u^S$  表示用户的增强视图对比学习损失函数,

$\text{sim}(\cdot)$  表示余弦相似度,  $\tau$  是一个温度超参数.  $\mathbf{e}_u^*, \mathbf{e}_u^*$  分别是用户  $u$  在扰动增强视图和 SVD 增强视图下的最终嵌入.

同理,我们可获得项目的损失  $L_i^S$ , 如式(8)所示:

$$L_i^S = - \sum_{i \in I} \ln \frac{\exp(\text{sim}(\mathbf{e}_i^*, \mathbf{e}_i^*)/\tau)}{\sum_{j \in I, j \neq i} \exp(\text{sim}(\mathbf{e}_i^*, \mathbf{e}_j^*)/\tau)}, \quad (8)$$

其中  $\mathbf{e}_i^*, \mathbf{e}_i^*$  分别是项目  $i$  在扰动增强视图和 SVD 增强视图下的最终嵌入.

综上,通过将 2 个损失函数相结合可得到增强视图对比学习目标函数,如式(9)所示:

$$L^S = L_u^S + L_i^S. \quad (9)$$

### 2.4 节点级对比学习

为探索不同层用户和候选项目间的关系并充分利用节点的语义信息,我们使用相似节点的语义嵌入构建对比学习,主要构造候选节点对比学习和候选结构邻居对比学习,以获得更高质量的节点嵌入表示.

#### 2.4.1 候选节点对比学习

对于给定的用户和候选项目,推荐的目标是预测它们之间交互的概率.根据协同过滤的基本原理,具有相似偏好的用户可能会与相同的项目进行交互,若用户  $u$  与候选项目  $i$  具有较高的交互可能性,则用户  $u$  在嵌入空间中会与候选项目  $i$  的历史交互用户相似,且用户  $u$  通常不与其他候选项目的历史交互用户相似.

**例 1.** 用户  $u_1$  喜欢吃汉堡包(候选项目),用户  $u_2$  曾经购买过汉堡包,用户  $u_3$  从来没有购买过汉堡包,那么我们认为在嵌入空间中,用户  $u_1$  和用户  $u_2$  的嵌入应拉近,用户  $u_1$  与用户  $u_3$  的嵌入则需拉远.

由例 1,在用户侧,我们将与候选项目  $i$  有关联的用户嵌入作为锚点(这里为  $\mathbf{e}_i^{(k)}$ ,  $i$  的奇数层节点为用户类型),将候选项目的一阶邻居节点  $u$  的嵌入(即  $\mathbf{e}_u^{(0)}$ )作为正实例,将其他项目的一阶邻居节点  $v$  的嵌入(即  $\mathbf{e}_v^{(0)}$ )作为负实例.用户侧候选节点对比学习损失函数表达式,如式(10)所示:

$$L_u^C = - \sum_{i \in I} \ln \frac{\exp(\text{sim}(\mathbf{e}_i^{(k)}, \mathbf{e}_u^{(0)})/\tau)}{\sum_{v \in U, v \neq u} \exp(\text{sim}(\mathbf{e}_i^{(k)}, \mathbf{e}_v^{(0)})/\tau)}, \quad (10)$$

其中  $L_u^C$  表示候选节点对比学习中用户侧的损失函数,  $\mathbf{e}_i^{(k)}$  表示第  $k$  层项目侧的嵌入,  $k$  为奇数,以保证锚点与正负实例为同一类型.

同理,在候选项目侧,用户  $u$  的历史交互项在嵌入空间中更靠近候选项目,而其他用户的历史交互项通常远离候选项目.

**例 2.** 用户  $u_1$  喜欢吃汉堡包(候选项目), 且他购买过爆米花, 用户  $u_2$  购买过蓝牙耳机, 则在嵌入空间中, 可认为汉堡包和爆米花的嵌入更接近, 爆米花和蓝牙耳机的嵌入一般较远.

由例 2, 我们使用与用户  $u$  有关联的项目嵌入(这里为  $e_u^{(k)}$ ,  $u$  的奇数层节点为项目类型)作为锚点, 该用户的一阶邻居节点  $i$  的嵌入(即  $e_i^{(0)}$ )作为正实例, 其他用户的一阶邻居节点  $j$  的嵌入(即  $e_j^{(0)}$ )作为负实例. 项目侧候选节点对比学习损失函数表达式, 如式(11)所示:

$$L_i^C = - \sum_{u \in U} \ln \frac{\exp(\text{sim}(e_u^{(k)}, e_i^{(0)})/\tau)}{\sum_{j \in I, j \neq i} \exp(\text{sim}(e_u^{(k)}, e_j^{(0)})/\tau)}, \quad (11)$$

其中  $L_i^C$  表示候选节点对比学习中项目侧的损失函数,  $e_u^{(k)}$  表示第  $k$  层用户侧的嵌入.

结合上述 2 种损失函数, 可得到候选节点对比学习的目标函数, 如式(12)所示:

$$L^C = \beta L_u^C + (1 - \beta) L_i^C, \quad (12)$$

其中  $\beta$  是控制 2 个损失函数强度的超参数.

#### 2.4.2 候选结构邻居对比学习

用户/项目的结构邻居包含相似兴趣的项目/用户. 用户的二阶邻居和项目的一阶邻居是相同类型的节点(均为用户类型). 在进行多层图卷积时, 它们将相互连接, 且存在长距离依赖关系. 若用户  $u$  与候选项目  $i$  交互, 用户  $u$  的结构邻居节点在嵌入空间中更接近候选项目  $i$  的相同类型的邻居节点.

**例 3.** 用户  $u_1$  喜欢吃汉堡包(候选项目), 且他购买过爆米花, 汉堡包与冰淇淋属于同一类型, 用户  $u_2$  购买过蓝牙耳机, 则在嵌入空间中爆米花( $u_1$  的结构邻居)与冰淇淋(候选项目的二阶邻居)的嵌入更接近, 一般与蓝牙耳机(其他用户的结构邻居)的嵌入更远.

由例 3, 我们选择目标用户的结构邻居节点  $i$  的嵌入(这里为  $e_i^{(k)}$ ,  $i$  的偶数层节点为项目类型)作为锚点, 用户  $u$  结构邻居的相同类型的邻居节点(项目类型)的嵌入(即  $e_u^{(k)}$ ,  $u$  的奇数层节点为项目类型)作为正实例, 其他用户  $v$  结构邻居的相同类型的节点(项目类型)的嵌入(即  $e_v^{(k)}$ ,  $v$  的奇数层节点为项目类型)作为负实例. 用户侧候选结构邻居对比学习损失函数表达式, 如式(13)所示:

$$L_u^P = - \sum_{i \in I} \ln \frac{\exp(\text{sim}(e_i^{(k)}, e_u^{(k)})/\tau)}{\sum_{v \in U, v \neq u} \exp(\text{sim}(e_i^{(k)}, e_v^{(k)})/\tau)}, \quad (13)$$

其中  $L_u^P$  表示候选结构邻居对比学习中用户侧的损失

函数,  $e_i^{(k)}$  表示第  $k$  层项目侧的嵌入,  $k$  为偶数, 以保证锚点与正负实例为同一类型.

同理, 在候选项目侧方面, 候选项目  $i$  的结构邻居节点也更接近用户  $u$  的相同类型的邻居节点.

**例 4.** 用户  $u_1$  喜欢汉堡包(候选项目), 用户  $u_2$  与  $u_1$  具有相似兴趣, 用户  $u_3$  曾经购买过汉堡包, 用户  $u_4$  未购买过汉堡包, 则在嵌入空间中  $u_2$  ( $u_1$  的二阶邻居)与  $u_3$  (候选项目的结构邻居)的嵌入更接近, 一般与  $u_4$  (其他项目的结构邻居)的嵌入更远.

由例 4, 我们使用候选项目的结构邻居节点  $u$  的嵌入(这里为  $e_u^{(k)}$ ,  $u$  的偶数层节点为用户类型)作为锚点, 候选项目  $i$  结构邻居的相同类型的邻居节点(用户类型)的嵌入(即  $e_i^{(k)}$ ,  $i$  的奇数层节点为用户类型)作为正实例, 其他项目  $j$  结构邻居的相同类型的邻居节点(用户类型)的嵌入(即  $e_j^{(k)}$ ,  $j$  的奇数层节点为用户类型)作为负实例. 项目侧候选结构邻居对比学习损失函数表达式, 如式(14)所示:

$$L_i^P = - \sum_{u \in U} \ln \frac{\exp(\text{sim}(e_u^{(k)}, e_i^{(k)})/\tau)}{\sum_{j \in I, j \neq i} \exp(\text{sim}(e_u^{(k)}, e_j^{(k)})/\tau)}, \quad (14)$$

其中  $L_i^P$  表示候选结构邻居对比学习中项目侧的损失函数,  $e_u^{(k)}$  表示第  $k$  层用户侧的嵌入.

结合以上 2 种损失函数, 得到候选结构邻居对比学习的目标函数, 如式(15)所示:

$$L^P = \beta L_u^P + (1 - \beta) L_i^P. \quad (15)$$

其中,  $\beta$  是控制 2 个损失函数强度的超参数.

#### 2.5 模型预测与推荐算法

下面给出本文模型 LA-MPGCL 的预测函数和最终优化的损失函数, 并详细阐述所提模型的整体推荐流程.

##### 2.5.1 模型预测

得到最终的节点嵌入(将扰动增强视图与 SVD 重构视图得到的节点嵌入相加)后, 采用内积来预测用户  $u$  和项目  $i$  交互的可能性, 如式(16)所示:

$$\hat{y}_{ui} = e_u^T e_i, \quad (16)$$

其中  $\hat{y}_{ui}$  是用户  $u$  与项目  $i$  交互的概率,  $e_u, e_i$  分别是用户  $u$  和项目  $i$  的最终嵌入.

与其他方法<sup>[9-11]</sup>类似, 本文模型采用多任务训练策略来联合优化推荐损失与对比学习损失. 推荐损失函数采用经典的贝叶斯个性化排名 (Bayesian personalized ranking, BPR) 方法, 该方法默认用户历史交互过项目的预测分数高于未交互过的项目, 如式(17)所示:

$$L^{\text{BPR}} = - \sum_{(u,i,j) \in O} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}), \quad (17)$$

其中 $\sigma$ 是 sigmoid 函数,  $O$ 为训练数据集,  $i$ 表示与用户  $u$  交互的项目,  $j$ 表示与用户  $u$  未交互的项目,  $\hat{y}_{ui}$ 表示用户  $u$  对交互项目  $i$  的预测分数,  $\hat{y}_{uj}$ 表示用户  $u$  对未交互项目  $j$  的预测分数。

通过优化 $L^{\text{BPR}}$ , 模型可有效地建模用户和项目间的交互。为进一步挖掘图结构信息和节点间信息, 我们将 2.3 节和 2.4 节中提出的 3 种对比学习损失函数与 $L^{\text{BPR}}$ 相结合, 得到联合优化的总损失函数, 如式(18)所示:

$$L^{\text{总}} = L^{\text{BPR}} + \lambda_1 L^{\text{S}} + \lambda_2 L^{\text{C}} + \lambda_3 L^{\text{P}} + \lambda_4 \|\Theta\|_2^2, \quad (18)$$

其中 $\lambda_1, \lambda_2, \lambda_3$ 分别是控制增强视图对比学习、候选节点对比学习和候选结构邻居对比学习损失权重的超参数,  $\lambda_4$ 是正则化系数,  $\Theta$ 表示模型的可训练参数,  $\|\Theta\|_2^2$ 表示 $L_2$ 正则化。

### 2.5.2 推荐算法

LA-MPGCL 模型通过构建在 2 个视角(视图级、节点级)下的 3 种对比学习(增强视图对比、候选节点对比、候选结构邻居对比), 达到提升模型泛化能力的目标, 并通过层注意力聚合节点嵌入, 以缓解过度平滑问题。具体过程见算法 1。

**算法 1.** 融合层注意力的多视角图对比学习推荐。

输入: 用户集  $U$ , 项目集  $I$ , 交互矩阵  $Y$ ;

输出: 推荐项目列表  $List$ 。

- ① 初始化用户  $u$  和项目  $i$  的嵌入  $e_u^{(0)}, e_i^{(0)}$ ; /\* 视图级对比学习 \*/
- ② 添加随机噪声构建  $u$  及  $i$  的扰动增强节点表示  $e_u^{(0*)}, e_i^{(0*)}$ ; /\* 见式(4) \*/
- ③ 利用 SVD 重构构建  $u$  及  $i$  的 SVD 增强节点表示  $e_u^{(0*)}, e_i^{(0*)}$ ; /\* 见式(5)(6) \*/
- ④ 通过层注意力机制聚合 2 个视图中  $u$  及  $i$  的节点表示  $e_u^*, e_i^*, e_u^*, e_i^*$ ; /\* 见式(2)(3) \*/
- ⑤ 获取增强视图对比学习损失函数  $L^{\text{S}}$ ; /\* 见式(7)~(9) \*/ /\* 节点级对比学习 \*/
- ⑥ 计算候选节点对比学习损失函数  $L^{\text{C}}$ ; /\* 见式(10)~(12) \*/
- ⑦ 计算候选结构邻居对比学习损失函数  $L^{\text{P}}$ ; /\* 见式(13)~(15) \*/ /\* 模型预测 \*/
- ⑧ 计算  $u$  与  $i$  交互的预测概率  $\hat{y}_{ui}$ ; /\* 见式(16) \*/
- ⑨ 计算联合训练损失函数  $L^{\text{总}}$  并更新参数; /\* 见式(17)(18) \*/
- ⑩ 根据  $\hat{y}_{ui}$  输出最终的推荐项目列表  $List$ 。

在算法 1 中, 行①得到用户  $u$  和项目  $i$  的初始化节点嵌入表示; 行②~⑤表示视图级对比学习操作过程, 通过添加随机噪声扰动与 SVD 重构分别构建扰动增强视图与 SVD 增强视图, 并利用层注意力机制聚合最终的节点嵌入表示, 通过 2 个增强视图的节点嵌入表示获取增强视图对比学习损失函数; 行⑥~⑦表示节点级对比学习过程, 利用节点语义性分别计算候选节点与候选结构邻居对比学习损失函数; 行⑧~⑩表示模型的预测过程, 利用前面得到的用户与项目嵌入表示计算预测概率, 并联合优化推荐损失与 3 个对比学习损失来更新参数, 最后根据预测概率获得最终的推荐列表。

## 3 实验设计与分析

我们针对以下问题进行实验以验证所提 LA-MPGCL 模型的有效性。

**问题 1.** 与经典的、最新的图卷积模型以及图对比学习模型相比, LA-MPGCL 模型表现如何。

针对该问题, 在 3.4 节展开了实验对比, 将我们的模型 LA-MPGCL 在 4 个数据集上与 3 个基于 GCN 的推荐模型, 以及 7 个基于图对比学习的推荐模型对比。在召回率、精确率、归一化折损累计增益这 3 个指标上, LA-MPGCL 均优于对比模型, 这主要归功于 LA-MPGCL 采取了多个视角下的对比学习方式, 提升其泛化能力。

**问题 2.** LA-MPGCL 模型的不同组件对其性能有何影响。是否具有必要性。

针对该问题, 在 3.5 节设置了消融实验, 重点研究了 LA-MPGCL 中的层注意力机制、随机扰动增强、SVD 重构增强、3 种对比学习方式对模型性能的影响, 共 6 个变体模型。实验结果表明, 变体模型的推荐效果均在不同程度上弱于 LA-MPGCL, 这表明各个组件是有效的、必要的。

**问题 3.** 超参数如何影响 LA-MPGCL 模型的性能。

针对该问题, 在 3.6 节设置了超参数的敏感度实验, 并重点选择控制随机扰动大小的参数、温度超参数、控制增强视图对比学习损失权重进行研究, 发现它们在不同的数据集上使模型达到最优值时的取值有所不同, 并分析了其潜在原因。

**问题 4.** LA-MPGCL 模型的时间复杂度如何。

针对该问题, 在 3.7 节进行了时间复杂度分析, 与最轻量的 LightGCN 模型、经典的图对比学习模型 SGL 以及最新的 AutoCF 模型进行对比, 分析 LA-

MPGCL 在时间复杂度上的表现。

**问题 5.** LA-MPGCL 模型在平衡过度平滑与过于均匀方面的表现如何。

针对该问题, 在 3.8 节对表征向量在空间中的嵌入分布情况展开可视化分析, 将 LA-MPGCL 与经典的非对比学习图模型 LightGCN 以及对比学习图模型 SGL 在 LastFM 数据集上进行对比, 阐述了 LA-MPGCL 在平衡过度平滑与过于均匀方面的优势。

### 3.1 数据集

采用 4 个公开数据集来验证模型的效果, 它们均是近些年被研究人员大量使用和研究的经典数据集, 分别是 LastFM, Gowalla, Ifashion, Yelp, 具体统计信息如表 2 所示。

Table 2 Information Statistics of Four Datasets

表 2 4 种数据集的信息统计表

数据集	用户数	项目数	交互数	稀疏度/%
LastFM	1 892	17 632	92 834	99.72
Gowalla	25 557	19 747	294 983	99.94
Ifashion	31 668	38 048	618 629	99.95
Yelp	29 601	24 734	1 517 326	99.79

1) LastFM<sup>①</sup>. 包含来自 Last.fm 网站在线音乐系统中用户的社交网络、标签和音乐艺术家收听信息, 而标签包括音乐播放数、音乐类型等。

2) Gowalla<sup>②</sup>. 来自于一个基于位置的社交网站, 用户通过签到共享其位置, 每个签到记录包括用户 ID、签到地点、时间等。

3) Ifashion<sup>③</sup>. 从阿里巴巴在线购物系统收集的时尚服装数据集, 每套服装中的物品均被视为向用户推荐的项目, 每个项目都包括类别和标题等属性。

4) Yelp<sup>④</sup>. 由 Yelp 网站发布包含互动和社会关系的商业评论数据集, 数据标签包括用户对商家的评分以及评论等。

### 3.2 实验环境与参数设置

本文模型 LA-MPGCL 采用 PyTorch 1.12 框架, 实验环境为: Windows10 64 位操作系统、PyCharm 2019、Python 3.9、内存 32 GB, CPU 为 AMD R7 3700 X 3.6 GHz, GPU 为 Nvidia GeForce RTX 2080Ti。

参照文献 [6, 31], 设置所提模型的嵌入向量维度为 64, 训练批量大小为 4 096, 学习率为 0.001, epoch

为 500, 控制节点级对比学习用户侧与项目侧强弱的超参数  $\beta$  为 0.5; 图卷积层数  $L$  在  $\{1, 2, 3, 4, 5\}$  上进行搜索, 最终设置  $L$  为 4 (因层数  $L$  小于 4 时效果较差, 而层数  $L$  为 4 或 5 时性能相差不大, 且层数  $L$  为 4 时, 模型已能学习绝大部分的有效信息, 故为减轻模型负担, 我们将层数  $L$  设置为 4); 正则化系数  $\lambda_4$  以及节点级对比学习的 2 个损失权重  $\lambda_2$  和  $\lambda_3$  均在  $\{1E-2, 1E-3, 1E-4, 1E-5, 1E-6\}$  上搜索, 最终均选择  $1E-4$ , 将 Adam 作为优化器优化模型, 使用 Xavier 作为网络参数初始化方法, 训练过程中采用早停技术防止过拟合, 推荐列表长度  $N$  设置为 10 或 50。

### 3.3 评价指标

训练模型后, 我们通过对训练集中所有未与用户交互的项目进行排名, 为每个用户生成个性化排名列表。为验证模型 LA-MPGCL 的有效性, 选取常用的召回率 (Recall)、精确率 (Precision)、归一化折损累计增益 (NDCG) 作为评价指标。

1) 召回率. 用来衡量用户所喜欢的物品中有多少是被推荐的, 如式 (19) 所示:

$$Recall = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|}, \quad (19)$$

其中  $R(u)$  表示推荐给用户  $u$  的项目列表,  $T(u)$  表示用户  $u$  实际选择的项目列表,  $R(u) \cap T(u)$  表示最终推荐列表中被用户  $u$  实际选择的项目列表。

2) 精确率. 用来衡量推荐结果中有多少推荐被用户所接受, 如式 (20) 所示:

$$Precision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|}. \quad (20)$$

3) 归一化折损累计增益. 对排序结果进行评价, 是折损累计增益 (DCG) 的改进方式. 因不同用户列表长度各不相同, 不同用户的 DCG 比较并不准确, 故需对不同用户的指标进行归一化. 使用 NDCG 可反映所推荐的项目是否在用户更加显眼的位置上, 从而表现预测的精准性, 如式 (21) 所示:

$$NDCG = \sum_{u \in U_{test}} \frac{1}{Y_u} \sum_{i=1}^k \frac{2^i - 1}{\lg(i + 1)}, \quad (21)$$

① <https://grouplens.org/datasets/hetrec-2011/>

② <https://snap.stanford.edu/data/loc-gowalla.html>

③ <https://github.com/wenyuer/POG>

④ <https://www.yelp.com/dataset>

其中  $U_{\text{test}}$  表示测试用户集,  $Y_u$  表示用户  $u$  的最大  $NDCG$  值;  $t_i = 1$  表示击中,  $t_i = 0$  表示未击中.

3.4 对比实验与分析 (针对问题 1)

本节将所提模型 LA-MPGCL 与当前较为流行的图卷积以及图对比学习模型进行对比分析, 阐述 LA-MPGCL 的优势.

3.4.1 对比模型

对比模型主要分成两大类: 1) 基于图卷积的推荐模型, 其中 NGCF<sup>[5]</sup> 是最先将 GCN 的技术应用在推荐中的模型, 而 LightGCN<sup>[6]</sup> 在其基础之上去掉了非激活函数与特征变换, 在更加轻便的同时, 性能也得到极大提升, 使其成为目前最受欢迎的图卷积模型之一. 而 IMP-GCN<sup>[14]</sup> 是基于 LightGCN 的改进, 通过划分子图并在子图内执行图卷积操作的方式提升性能. 2) 基于图对比学习的推荐模型, 其中 SGL<sup>[9]</sup> 是最先将数据增强及图对比学习引入推荐中的模型, 而 SimGCL<sup>[11]</sup> 认为 SGL 中的图增强方式效果有限, 探究了图对比学习效果变化的关键因素, 并提出通过随机噪声简化图增强过程. HCCF<sup>[12]</sup> 和 NCL<sup>[13]</sup> 设计基于启发式的策略来构建用于嵌入对比的视图. LightGCL<sup>[30]</sup> 和 AutoCF<sup>[31]</sup> 则是近两年提出的探索数据增强方式的模型. 将 LA-MPGCL 与上述模型进行对比, 可验证 LA-MPGCL 的有效性.

① NGCF. 这是一种基于图的 CF 方法, 主要遵循标准 GCN, 在消息传递期间将二阶特征交互编码到消息中.

② LightGCN. 采用简化的 GCN 结构, 在 NGCF 基础上去掉非线性激活函数与特征变换.

③ IMP-GCN. 兴趣感知消息传递 GCN 推荐模型, 构建包含用户和项目信息的子图, 并在子图内执行高阶图卷积, 可有效地识别用户的核心兴趣.

④ SGL-ED/ND. 通过数据增强应用辅助的自监督对比学习任务, 即边丢弃 (ED) 或节点丢弃 (ND), 来增强使用 GNN 的用户/项目表示学习.

⑤ SimGCL. 通过直接向特征表示中注入随机噪声来简化对比学习的图增强过程.

⑥ HCCF. 用超图编码全局图信息, 并将其与用 GCN 编码的局部信息进行对比.

⑦ NCL. 通过增强结构和语义相关的训练来增强图的对比学习.

⑧ LightGCL. 一个轻量级的图对比学习框架, 利用矩阵分解生成增强视图嵌入对比.

⑨ AutoCF. 一种自动化的自监督数据增强方法, 可通过遮蔽图自编码器的方式增强基于图神经网络的协同过滤.

3.4.2 对比结果与分析

我们与相关经典的、最新的模型在 4 个数据集上进行详细对比. 参照文献 [9], LA-MPGCL 以 7 : 1 : 2 的比例划分训练集、验证集与测试集, 实验具体结果如表 3、表 4 所示.

图 3 是表 3、表 4 中实验结果更直观的表现, 图 3(a)~(d) 分别表示在数据集 LastFM, Gowalla,

Table 3 Performance Comparison on LastFM and Gowalla Datasets

表 3 LastFM 和 Gowalla 数据集上的性能对比

模型	LastFM						Gowalla					
	R@10	R@50	P@10	P@50	N@10	N@50	R@10	R@50	P@10	P@50	N@10	N@50
NGCF	0.137 9	0.323 6	0.069 4	0.032 5	0.123 7	0.190 5	0.143 5	0.328 7	0.050 3	0.024 6	0.115 6	0.168 1
LightGCN	0.150 1	0.342 8	0.075 2	0.034 5	0.134 0	0.203 9	0.166 0	0.379 4	0.059 4	0.029 0	0.133 5	0.194 2
IMP-GCN	0.151 9	0.343 5	0.075 3	0.034 5	0.129 9	0.199 5	0.170 2	0.375 3	0.058 6	0.027 9	0.137 2	0.195 8
SGL-ED	0.170 2	0.373 3	0.084 8	0.037 2	0.148 8	0.221 7	0.178 2	0.387 2	0.063 6	0.030 0	0.143 8	0.203 7
SGL-ND	0.172 3	0.373 4	0.085 8	0.037 4	0.153 5	0.226 5	0.169 7	0.357 7	0.061 8	0.028 5	0.139 1	0.193 1
SimGCL	0.169 2	0.370 7	0.084 5	0.037 2	0.150 7	0.223 8	0.170 5	0.386 4	0.061 8	0.029 9	0.138 6	0.200 1
HCCF	0.166 4	0.363 3	0.082 0	0.036 1	0.146 7	0.218 4	0.172 9	0.369 3	0.061 1	0.028 7	0.138 6	0.195 2
NCL	0.173 9	<u>0.380 0</u>	0.086 1	<u>0.037 9</u>	0.153 9	<u>0.228 9</u>	0.177 3	0.392 2	0.063 3	0.030 1	0.142 7	0.204 0
LightGCL	0.170 5	0.367 6	0.084 9	0.036 8	0.151 8	0.223 2	0.174 5	0.384 7	0.061 6	0.029 8	0.139 3	0.199 9
AutoCF	<u>0.176 9</u>	0.364 6	<u>0.086 7</u>	0.036 3	<u>0.155 2</u>	0.223 9	<u>0.178 8</u>	<u>0.395 6</u>	<u>0.064 1</u>	<u>0.030 6</u>	<u>0.144 6</u>	<u>0.206 6</u>
LA-MPGCL (本文)	<b>0.181 1</b>	<b>0.383 3</b>	<b>0.090 2</b>	<b>0.038 2</b>	<b>0.162 4</b>	<b>0.235 7</b>	<b>0.188 8</b>	<b>0.400 2</b>	<b>0.067 9</b>	<b>0.031 5</b>	<b>0.153 6</b>	<b>0.214 3</b>
性能提升/%	2.37	0.87	4.04	0.79	4.64	2.97	5.59	1.16	5.93	2.94	6.22	3.73

注: 黑体值为本文模型 LA-MPGCL 的指标值, 对比模型的最优值用下划线标示. Recall, Precision, NDCG 分别简称为 R, P, N. 性能提升是 LA-MPGCL 相对于最优值的性能提升情况. 例如, 就 LastFM 的 R@10 指标, 在对比模型中, AutoCF 表现最优, LA-MPGCL 与之相比, 提升了 2.37%.

Table 4 Performance Comparison on Ifashion and Yelp Datasets

表 4 Ifashion 和 Yelp 数据集上的性能对比

模型	Ifashion						Yelp					
	R@10	R@50	P@10	P@50	N@10	N@50	R@10	R@50	P@10	P@50	N@10	N@50
NGCF	0.039 3	0.127 7	0.025 1	0.016 7	0.037 0	0.066 8	0.047 2	0.152 2	0.045 5	0.029 9	0.057 2	0.093 7
LightGCN	0.044 3	0.139 4	0.027 5	0.018 1	0.041 4	0.073 6	0.050 5	0.160 1	0.047 8	0.030 9	0.061 0	0.099 1
IMP-GCN	0.047 3	0.147 6	0.030 4	0.019 3	0.045 6	0.079 2	0.051 0	0.160 9	0.048 6	0.031 6	0.061 9	0.100 2
SGL-ED	0.048 1	0.146 0	0.030 0	0.019 1	0.045 3	0.078 3	0.052 7	0.161 7	0.050 2	0.031 2	0.064 7	0.101 9
SGL-ND	0.045 4	0.131 1	0.028 9	0.017 5	0.043 2	0.071 9	0.057 7	0.174 4	0.054 1	0.033 7	0.069 8	0.110 2
SimGCL	0.050 0	<u>0.154 6</u>	0.031 4	<u>0.020 2</u>	0.047 2	0.082 3	0.054 2	0.169 0	0.052 2	0.033 2	0.066 3	0.105 8
HCCF	0.045 6	0.141 6	0.029 3	0.018 9	0.043 5	0.075 8	0.059 0	0.178 3	0.055 9	0.034 8	0.071 1	0.112 1
NCL	0.048 6	0.151 7	0.030 7	0.019 7	0.045 8	0.080 4	0.055 5	0.174 3	0.052 3	0.033 7	0.066 4	0.107 8
LightGCL	0.049 3	0.148 3	0.031 0	0.019 6	0.046 8	0.080 2	0.058 4	0.178 5	0.055 3	0.034 6	0.070 8	0.112 0
AutoCF	<u>0.050 2</u>	0.154 1	<u>0.031 8</u>	0.020 1	<u>0.047 5</u>	<u>0.082 4</u>	<u>0.064 2</u>	<u>0.190 3</u>	<u>0.060 2</u>	<u>0.037 0</u>	<u>0.077 1</u>	<u>0.120 3</u>
<b>LA-MPGCL(本文)</b>	<b>0.054 2</b>	<b>0.162 3</b>	<b>0.033 7</b>	<b>0.021 1</b>	<b>0.050 9</b>	<b>0.087 1</b>	<b>0.064 4</b>	<b>0.193 5</b>	<b>0.060 7</b>	<b>0.037 3</b>	<b>0.077 7</b>	<b>0.121 8</b>
性能提升/%	7.97	4.98	5.97	4.46	7.16	5.70	0.31	1.68	0.83	0.81	0.78	1.25

注：黑体值为本文模型 LA-MPGCL 的指标值，对比模型的最优值用下划线标示。Recall, Precision, NDCG 分别简称为 R, P, N。性能提升是 LA-MPGCL 相对于最优值的性能提升情况。例如，就 Ifashion 的 R@10 指标，在对比模型中，AutoCF 表现最优，LA-MPGCL 与之相比，提升了 7.97%。

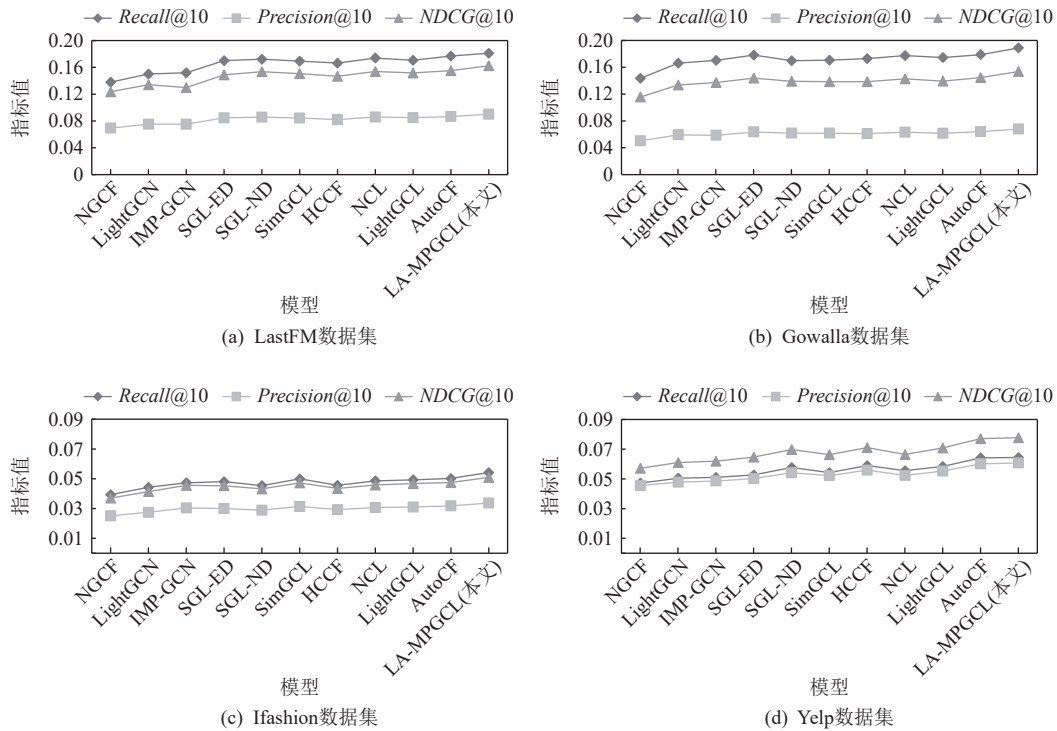


Fig. 3 Comparison on each metric of the models on four datasets

图 3 各模型在 4 个数据集上的各指标对比

Ifashion, Yelp 的结果。为方便展示，我们仅选取  $N=10$  时 3 个指标的变化情况，即每个子图中的 3 条曲线分别对应  $Recall@10$ ,  $Precision@10$ ,  $NDCG@10$  的指标值。

由表 3、表 4 与图 3 可看出，LA-MPGCL 在 3 个指标  $Recall@N$ ,  $Precision@N$ ,  $NDCG@N$  ( $N$  取 10 或 50) 上，均优于对比模型。下面是具体分析，其中“性能

提升”的计算，以对比模型 NGCF 的  $Recall@N$  指标为例，LA-MPGCL 相对于模型 NGCF 在 4 个数据集的  $Recall@N$  ( $N$  取 10 或 50) 指标上的 8 个提升比例中选取最小值 18.45%。

1) 与基于 GCN 的模型相比。① NGCF 遵循标准的 GCN 结构，标准 GCN 中一些“无用”的模块成为模型的负担导致模型效果一般。相较于 NGCF 模型，

LA-MPGCL 在  $Recall@N$ ,  $Precision@N$ ,  $NDCG@N$  指标上分别最少提升 18.45%, 17.54%, 23.73%。② LightGCN 在 NGCF 的基础上舍弃了非线性激活函数和特征变换, 使模型更加轻便的同时达到了更优的效果, 但并未逃脱 GCN 模型过度平滑的限制。相较于 LightGCN, LA-MPGCL 在  $Recall@N$ ,  $Precision@N$ ,  $NDCG@N$  指标上分别最少提升 5.48%, 8.62%, 10.35%。③ IMP-GCN 通过对具有相似兴趣的用户划分为统一子图, 在子图内执行图卷积, 使得模型可更深层次地挖掘信息, 但这同时也增加了模型训练时长。相较于 IMP-GCN, LA-MPGCL 在  $Recall@N$ ,  $Precision@N$ ,  $NDCG@N$  指标上分别最少提升 6.63%, 9.33%, 9.45%。

基于图卷积的模型大多都关注于更好地聚合高阶协同信息, 但推荐系统本身存在的数据稀疏性问题会限制模型效果的进一步提升。因此, LA-MPGCL 采用多种图数据增强和对比学习的方式缓解该问题, 并在最终聚合节点嵌入时采用层注意力机制更好地利用图结构的高阶连通性, 这是优于基于 GCN 的重要原因。

2) 与基于图信息随机丢弃的 GCL 模型相比。① SGL-ED 通过随机丢弃边构造增强视图进行对比学习。相较于 SGL-ED, LA-MPGCL 在  $Recall@N$ ,  $Precision@N$ ,  $NDCG@N$  指标上分别最少提升 2.68%, 2.69%, 5.20%。② SGL-ND 利用随机丢弃节点构建增强视图进行对比学习。相较于 SGL-ND, LA-MPGCL 在  $Recall@N$ ,  $Precision@N$ ,  $NDCG@N$  指标上分别最少提升 2.65%, 2.14%, 4.06%。③ 进一步观察, SGL-ED 在 Gowalla 和 Ifashion 数据集上优于 SGL-ND, 而 SGL-ND 在 LastFM 和 Yelp 数据集上表现更佳。其原因是 SGL-ND 在丢弃节点的同时也会丢弃周围的一些边, 因此, 高度节点丢弃可能会显著改变原有的图结构, 从而影响信息聚集。在数据相对稠密的 LastFM 和 Yelp 数据集上这种影响会小, 而在数据更稀疏的 Gowalla 和 Ifashion 数据集上则会造成更坏的影响导致效果不佳。

边丢弃和节点丢弃的增强方式, 一方面需人工干预选择更优的增强方式, 另一方面一般会破坏原有图结构, 影响模型性能。而 LA-MPGCL 摒弃了这一方式, 采用随机扰动和 SVD 重构的增强方式, 可更好地提取有效信息。

3) 与基于随机噪声及启发式策略的 GCL 模型相比。① SimGCL 放弃了基于图信息随机丢弃的图增强方式, 而是将随机均匀噪声添加到原始表示中用于表征级数据增强, 添加不同程度的随机噪声会在对比视图之间产生不同方差, 由于其噪声大小可控, 所

以仍然保留原始信息。与随机丢弃的增强方式相比, 这种方式可使嵌入空间分布更均匀。相较于 SimGCL, LA-MPGCL 在  $Recall@N$ ,  $Precision@N$ ,  $NDCG@N$  指标上分别最少提升 3.40%, 2.69%, 5.32%。② HCCF 通过构建超图捕获的全局依赖信息与 GCN 获取的局部信息进行对比学习。相较于 HCCF, LA-MPGCL 在  $Recall@N$ ,  $Precision@N$ ,  $NDCG@N$  指标上分别最少提升 5.51%, 5.82%, 7.92%。③ NCL 利用聚类获取相似用户和项目的节点嵌入, 并以此构建对比学习。相较于 NCL, LA-MPGCL 在  $Recall@N$ ,  $Precision@N$ ,  $NDCG@N$  指标上分别最少提升 0.87%, 0.79%, 2.97%。

这些方法虽然在一定程度上有效, 但均仅采用单一的对比学习方式, 模型的泛化能力不足, 很难适应不同的推荐任务。例如, NCL 在 LastFM 数据集上有 3 个指标达到了对比模型中最好的效果, 但在 Yelp 数据集上表现不佳。LA-MPGCL 在视图级和节点级 2 个视角上构建了 3 种对比学习, 在保留原有图结构的同时, 充分挖掘有效信息, 极大地提升了模型的泛化能力, 使模型在各个数据集上均有最优的表现。

4) 与近期的 GCL 模型相比。① LightGCL 通过矩阵分解重构增强图的方式提取图结构中的重要信息。相较于 LightGCL, LA-MPGCL 在  $Recall@N$ ,  $Precision@N$ ,  $NDCG@N$  指标上分别最少提升 4.03%, 3.80%, 5.60%。② AutoCF 通过自适应遮蔽图自编码器的方式增强基于图神经网络的协同过滤, 进而达到提高模型泛化能力的目标, 其在 4 个数据集上均有较为优异的表现。相较于 AutoCF, LA-MPGCL 在  $Recall@N$ ,  $Precision@N$ ,  $NDCG@N$  指标上分别最少提升 0.31%, 0.81%, 0.78%。

与这些模型不同的是, LA-MPGCL 不但考虑了视图级的增强图对比学习方式, 还考虑了用户和候选项目间的节点级关系, 使模型在泛化能力上具备更强的优势, 层注意力机制的引入也使模型训练更加稳定。

综合以上分析, LA-MPGCL 在 LastFM, Gowalla, Ifashion, Yelp 这 4 个数据集上与最优的对比模型相比,  $Recall@N$  值最少提升 0.87%, 1.16%, 4.98%, 0.31%;  $Precision@N$  值最少提升 0.79%, 2.94%, 4.46%, 0.81%;  $NDCG@N$  值最少提升 2.97%, 3.73%, 5.70%, 0.78%。从对比结果来看, LA-MPGCL 在 Gowalla 和 Ifashion 数据集的提升更多, 原因在于这 2 个数据集的数据更为稀疏, LA-MPGCL 一方面通过视图级对比学习提取图结构有效信息, 另一方面探索用户与候选项目间的关系, 丰富了节点语义信息, 从而使模型对更为稀疏的数据产生的推荐效果更好。可见, LA-MPGCL

在数据集较稀疏的情况下能有更好的表现。

LA-MPGCL 在各数据集的各项指标上均达到最优效果, 这主要归功于: 1) 摒弃了随机丢弃的图增强方式, 采用随机扰动和 SVD 重构的增强方式进行视图级对比学习, 在更有效地提取图结构信息的同时避免了对图结构的破坏。2) 在考虑图结构增强的同时, 考虑了用户和候选项目间的节点级对比学习, 使模型的泛化能力更强。3) 在最终节点嵌入聚合时, 采用层注意力机制更有效地利用图结构的高阶连通性, 学习更高质量的节点嵌入表示。

### 3.5 消融实验与分析 (针对问题 2)

为阐明 LA-MPGCL 中层注意力机制、随机扰动增强、SVD 重构增强这 3 种对比学习的必要性和有效性, 我们对模型做消融实验对比, 构建的 6 种变体模型分别用变体模型 1、变体模型 2、变体模型 3、变体模型 4、变体模型 5、变体模型 6 表示。1) 变体模型 1 是在模型中去掉了层注意力机制聚合函数, 改为用经典的平均池化聚合最终节点嵌入(其内容见 2.2 节); 2) 变体模型 2 是在扰动增强视图上去掉了随机扰动的增强方式, 采用原始图结构信息(其内容见 2.3.1 节); 3) 变体模型 3 是在 SVD 增强视图上去掉了 SVD 重构的增强方式, 使用原始图结构信息(其内容见 2.3.2 节); 4) 变体模型 4 是在模型中去掉了增强视图对比学习(其内容见 2.3.3 节); 5) 变体模型 5 是在模型中去掉了节点级对比学习中的候选节点对比学习(其内容见 2.4.1 节); 6) 变体模型 6 是在模型中

去掉了节点级对比学习中的候选结构邻居对比学习(其内容见 2.4.2 节)。

具体地, 变体模型 1 主要是验证层注意力机制聚合对比平均池化聚合的优势, 变体模型 2 和变体模型 3 主要是验证各自图增强方式的有效性, 变体模型 4、变体模型 5、变体模型 6 主要是验证 3 种对比学习的合理性。表 5 是 6 种变体模型的组件构成详细情况。

Table 5 Component Composition of Several Variant Models

表 5 多个变体模型组件构成情况

变体模型	层注意力	随机扰动	SVD 重构	$L^S$	$L^C$	$L^P$
变体模型 1	×	○	○	○	○	○
变体模型 2	○	×	○	○	○	○
变体模型 3	○	○	×	○	○	○
变体模型 4	○	○	○	×	○	○
变体模型 5	○	○	○	○	×	○
变体模型 6	○	○	○	○	○	×
LA-MPGCL (本文)	○	○	○	○	○	○

注: “×”表示不存在该组件; “○”表示存在该组件。

我们选取  $Recall@10$  和  $NDCG@10$  指标进行消融实验对比, 如图 4 所示。

在图 4 中, 为更明显地展示各变体模型与 LA-MPGCL 性能上的差异, 图 4(a)(c) 的纵坐标分别从 0.14 和 0.04 始。从图 4 中可看出, SVD 重构对实验结

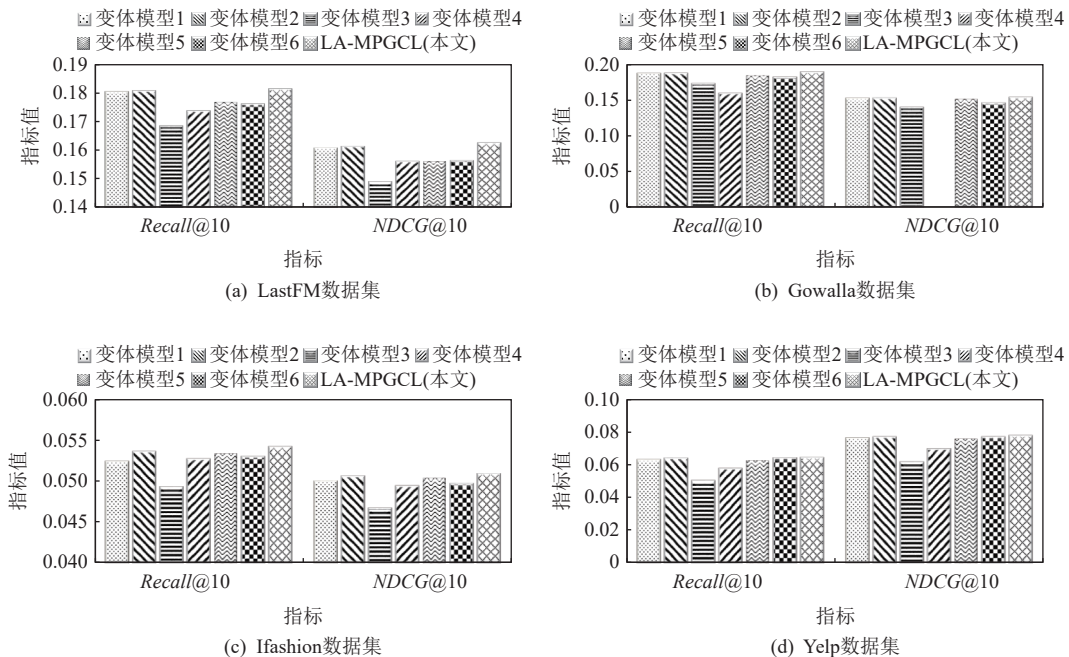


Fig. 4 Comparison of  $Recall@10$  and  $NDCG@10$  between variant models and proposed LA-MPGCL

图 4 变体模型与本文模型 LA-MPGCL 的  $Recall@10$  和  $NDCG@10$  对比

果的影响最为明显,不使用SVD重构增强的变体模型3在各指标上明显低于LA-MPGCL,也低于使用了SVD重构增强的变体模型1、变体模型2、变体模型5、变体模型6,3种对比学习辅助任务均会导致模型结果产生较大变化,此外,层注意力机制和随机扰动增强对模型性能均有不同程度的影响.具体分析如下:

1)SVD重构对模型效果的平均影响最大.在LastFM, Ifashion, Yelp中没有SVD重构的变体模型3效果最差,在Gowalla中没有SVD重构的变体模型也仅高于没有视图级对比学习的变体模型4.SVD重构可提取图结构中的有效信息,进而构建出合理的增强视图,提升模型的性能.此外,与变体模型4相比,变体模型3还多了扰动增强视图与原始图的对比,但变体模型3在LastFM, Ifashion, Yelp这3个数据集上的性能反而低于变体模型4,我们认为去除了SVD增强之后,视图级对比已产生较大变化,辅助任务和推荐任务的关系发生了改变,这时需调整超参数使两者的关系重新达到平衡状态.从实验结果看,SVD重构有助于模型学习更高质量的节点嵌入,进而提高模型性能.

2)3种对比学习辅助任务对模型有较大影响.分别去除了3个对比学习损失函数的变体模型4、变体模型5、变体模型6的各指标值均低于LA-MPGCL,说明这3个辅助任务均对模型效果有帮助.这3个变体模型互相对比,变体模型4的效果最差,说明视图级对比学习对模型的影响最大.而由图4(b)(d)可知,候选节点对比学习和候选结构邻居对比学习在不同的数据集上影响有所不同,这是由于数据集之间的节点表示和节点关系不同造成的.可肯定的是,这3个辅助任务均有助于提升模型性能,这也是LA-MPGCL泛化能力强的重要原因.

3)层注意力机制和扰动增强对模型性能有一定影响.没有层注意力机制的变体模型1和没有扰动增强的变体模型2性能均弱于LA-MPGCL.层注意力关注每一层权重,更好地捕捉层间的有效信息,从而提升模型高阶连通性,缓解过度平滑问题.但相较于节点数、边数来说,层数一般非常小,因此层注意力机制对模型性能的提升相对有限.随机扰动可使节点嵌入分布更加均匀,提升模型泛化能力.然而,由于SVD重构已提取重要的有效信息,因此随机扰动的增强方式仅能在一定程度上提高嵌入质量.

综上分析,模型的各个组件对提升模型的效果均有一定程度上的作用.1)在与变体模型3对比分析

时发现,SVD重构的影响较大,可从图中提取有效的信息特征,使模型能提高节点嵌入的质量.2)在与变体模型4、变体模型5、变体模型6对比分析后,可得出3种对比学习均有助于改善模型效果,提升模型的泛化能力.3)在与变体模型1和变体模型2对比分析中发现,层注意力机制可稳定深层GCN的训练,从而缓解过度平滑问题,而随机扰动增强可使模型表征分布更均匀.

### 3.6 参数敏感度分析(针对问题3)

这里对3个关键超参数:控制随机扰动大小的参数 $\varepsilon$ 、温度超参数 $\tau$ 和控制增强视图对比学习损失权重的超参数 $\lambda_1$ 进行实验分析,阐述它们在不同数据集上对模型性能的影响.

#### 1) 超参数 $\varepsilon$ 的影响

LA-MPGCL构建了扰动增强视图,通过式(4)约束条件中的参数 $\varepsilon$ 控制随机扰动大小,其值越大,则扰动的影响越大.越大的扰动会使节点表示与原始表示偏离越远,从而影响推荐效果.过小的扰动又不足以对节点表示产生影响,极端情况下,扰动增强视图将退化为原始图.为选取最优的 $\varepsilon$ 值,实验设置 $\varepsilon \in \{0.001, 0.01, 0.1, 0.2, 0.5\}$ .不同数据集上各指标值随 $\varepsilon$ 变化情况如图5所示.

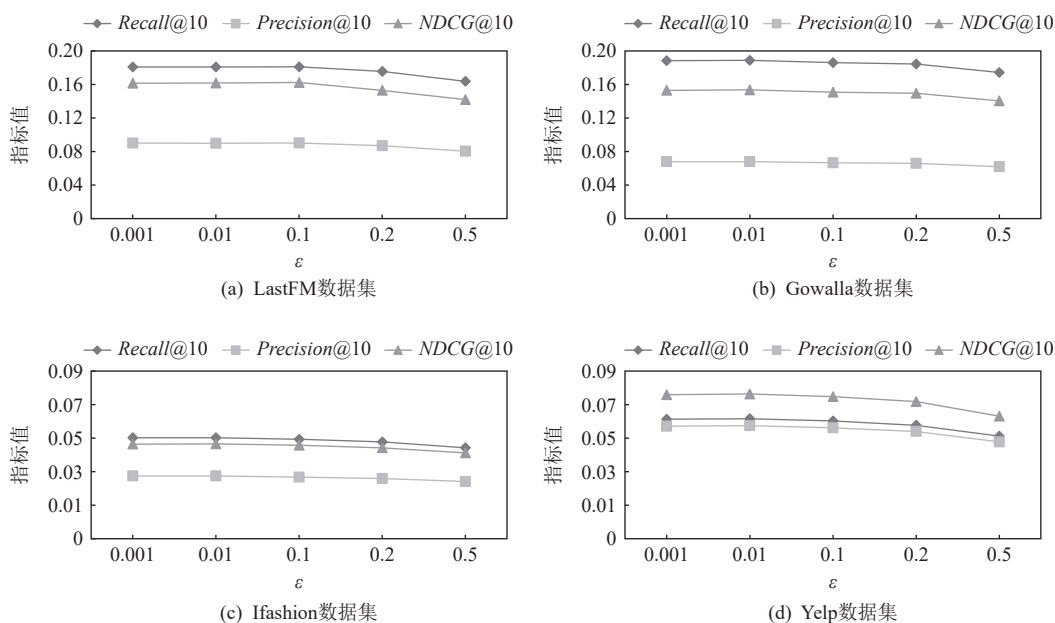
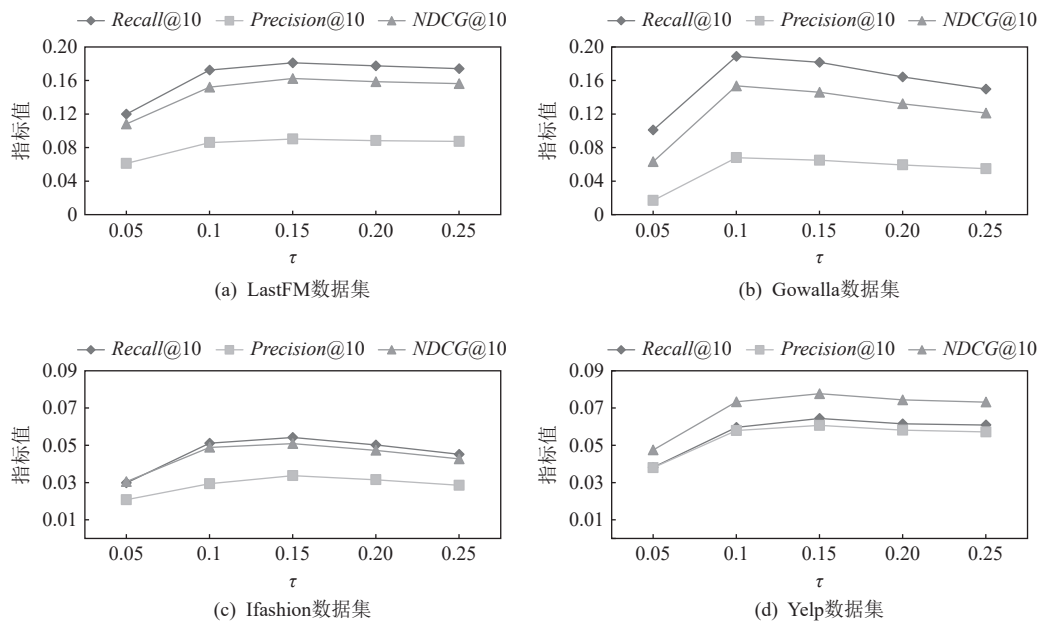
从图5可看出,添加过多的扰动会使模型性能下降,当 $\varepsilon > 0.1$ 时,模型在4个数据集上的表现均有所下降.在LastFM中, $\varepsilon = 0.1$ 时,模型的性能指标达到最大值;在Gowalla, Ifashion, Yelp中,模型性能在 $\varepsilon = 0.01$ 时达到最优.

可见,在不同的数据集上,参数 $\varepsilon$ 的最优取值有所不同,我们认为这是合理的.因为每个数据集的原始嵌入分布不同,只有当扰动的大小合理时,才能使特征分布更加均匀.

#### 2) 超参数 $\tau$ 的影响

在对比学习中, $\tau$ 是非常重要的温度超参数,其作用是调节模型对困难负样本的重视程度. $\tau$ 越大,重视程度越低; $\tau$ 越小,重视程度越高.当 $\tau$ 很小的时候,会强制把附近所有的其他样本推开,这样难免伤及无辜,一些原来比较相似的样本也会被拉远,导致模型语义结构较差.而当 $\tau$ 增至无穷大时,对比学习对所有负样本一视同仁,也就失去了其意义.为选取最优的 $\tau$ 值,实验设置 $\tau \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$ .不同数据集上各指标值随 $\tau$ 变化情况如图6所示.

从图6可看出,随着 $\tau$ 值的增大,模型性能先增后减.在Gowalla数据集中, $\tau = 0.1$ 时,模型性能达到最佳;在LastFM, Ifashion, Yelp中, $\tau = 0.15$ 时,模型获

Fig. 5 The influence of parameter  $\varepsilon$ 图5 参数 $\varepsilon$ 的影响Fig. 6 The influence of parameter  $\tau$ 图6 参数 $\tau$ 的影响

得最佳效果。

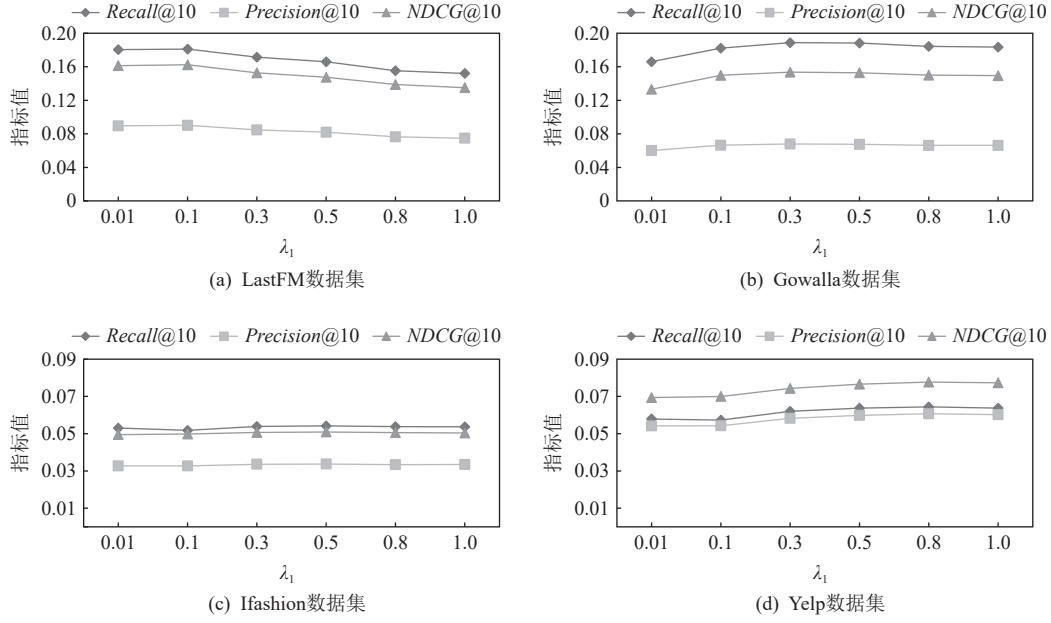
可见,在不同的数据集上,参数 $\tau$ 的最优取值有所不同,我们认为这是合理的.因不同数据集的正负样本分布不同,故需设置合适的 $\tau$ 值使模型的代表分布的均匀性更加合理,以达到较优性能。

### 3) 超参数 $\lambda_1$ 的影响

$\lambda_1$ 是控制增强视图对比学习辅助任务占联合学习任务比重的参数,其值越大,增强视图对比学习对模型性能的影响越大.同样是对比学习损失权重的

参数 $\lambda_2$ 和 $\lambda_3$ ,因其原理和调优过程与 $\lambda_1$ 相似,故在3.2节直接给出其取值.为选取最优的 $\lambda_1$ 值,实验设置 $\lambda_1 \in \{0.01, 0.1, 0.3, 0.5, 0.8, 1.0\}$ .不同数据集下各指标值随 $\lambda_1$ 变化情况如图7所示。

从图7可看出,在LastFM中, $\lambda_1 = 0.1$ 时,模型性能达到最优,之后随着取值的增大而逐渐降低;在Gowalla中, $\lambda_1 = 0.3$ 时,各指标达到最大值;在Ifashion中, $\lambda_1 = 0.5$ 时,模型效果达到最佳;在Yelp中, $\lambda_1 = 0.8$ 时,模型表现最优。

Fig. 7 The influence of parameter  $\lambda_1$ 图7 参数 $\lambda_1$ 的影响

可见,在不同的数据集上,参数 $\lambda_1$ 的最优取值有所不同,我们认为这是合理的.在不同的数据集中,原始数据质量并不相同,恰当的 $\lambda_1$ 取值能更好地平衡增强视图对比学习任务和其他任务的关系,从而达到更佳的推荐效果.

综上对各主要超参数的分析可知,在 LastFM, Gowalla, Ifashion, Yelp 数据集中,我们设置 $\epsilon$ 取值分别为 0.1, 0.01, 0.01, 0.01,  $\tau$ 取值分别为 0.15, 0.1, 0.15,  $\lambda_1$ 取值分别为 0.1, 0.3, 0.5, 0.8 时,模型整体可达到最优效果.

### 3.7 时间复杂度分析 (针对问题 4)

与现有图卷积以及图对比模型相比, LA-MPGCL 构建了 2 个视角下的 3 种对比学习,并在聚合最终嵌入时采用层注意力机制,在性能上得到一定提升,但其在时间复杂度上并不总占优.

1) 就 LA-MPGCL 而言,扰动增强部分的时间复杂度为  $O(|V|d)$ , 其中  $|V|$  表示用户和项目总数,  $d$  表示嵌入维度; SVD 重构增强部分的时间复杂度为  $O(q|E|)$ , 其中  $q$  是 SVD 分解所需的秩,  $|E|$  为边数; 视图级对比学习模块的时间复杂度为  $O(2|E|Ld+2q|V|Ld)$ , 其中  $L$  为层数; 节点级对比学习模块的时间复杂度为  $O(2|V|Ld)$ ; 模型训练的时间复杂度为  $O(3Bd+B Md)$ , 其中  $B$  为批量大小,  $M$  为一个批次中的节点数. 因  $q \ll Ld$ , 故 LA-MPGCL 的时间复杂度大致为  $O(|E|Ld+q|V|Ld+B Md)$ .

2) 轻量级的图卷积模型 LightGCN 的运算主要体现在执行图卷积操作过程中, 时间复杂度为  $O(2|E|Ld)$ ,

其模型训练的时间复杂度为  $O(2Bd)$ , 因此该模型时间复杂度大致为  $O(|E|Ld+Bd)$ . LA-MPGCL 与其相比复杂度有所增加, 但换来了较大的性能提升, 这是值得的.

3) 运用图增强的图对比学习模型 SGL (包括 SGL-ED 与 SGL-ND) 的运算主要体现在图增强和图卷积操作中, 这里以 SGL-ED 为例, 时间复杂度为  $O(2|E|Ld+4\rho|E|Ld)$ , 其中  $\rho$  为随机丢弃比例, 一般是 0 到 1 之间的数, 模型训练时间复杂度为  $O(3Bd+B Md)$ , 因此该模型时间复杂度大致为  $O(|E|Ld+B Md)$ . LA-MPGCL 与其相比复杂度略有增加, 但性能提升可观, 这点牺牲也是值得的.

4) 平均性能最好的对比模型 AutoCF 的运算主要体现在遮蔽范式学习和遮蔽图自编码器组件中, 其时间复杂度大致为  $O(k|E|d+|E||V_k|+|E|d^2)$ , 其中  $|V_k|$  表示第  $k$  次迭代的中心节点数,  $k$  是超参数, 一般  $k \ll d$ . 模型训练的时间复杂度为  $O(3Bd+B Md)$ , 因此该模型的时间复杂度大致为  $O(|E||V_k|+|E|d^2+B Md)$ . 由于通常  $L < d$ , 且  $q|V|Ld < |E||V_k|$ , 因此 LA-MPGCL 在大多数情况下的复杂度低于 AutoCF, 另外通过对比实验已知, 其在性能上也优于该模型.

为直观感受各模型效率上的差别, 本文选取了 2 个较大的数据集 Ifashion 和 Yelp 运行 10 次迭代取平均的平均运行时间作对比, 具体如表 6 所示.

综上分析, 各模型时间复杂度高低为 AutoCF > LA-MPGCL > SGL-ED > LightGCN, 这与表 6 的运行结

**Table 6 Comparison of Running Time of Each Model Under One Epoch**

表 6 各模型 1 次迭代下的运行时间对比

数据集	模型	运行时间/s
Ifashion	LightGCN	12
	SGL-ED	27
	AutoCF	403
	LA-MPGCL (本文)	42
Yelp	LightGCN	22
	SGL-ED	50
	AutoCF	783
	LA-MPGCL (本文)	66

果一致. 尽管 LA-MPGCL 的复杂度略高于 LightGCN 和 SGL-ED, 但整体上相差不大, 对于目前大多数 GPU/CPU 而言, 这是可接受的. 在更大的数据集 Yelp 上, LA-MPGCL 的运行时间增长趋势明显低于对比模型, 且就性能而言, LA-MPGCL 优于 LightGCN 和 SGL-ED. 相较于 LightGCN, LA-MPGCL 在  $Recall@N$ ,  $Precision@N$ ,  $NDCG@N$  指标上分别最少提升 5.48%, 8.62%, 10.35%. 相较于 SGL-ED, LA-MPGCL 在  $Recall@N$ ,  $Precision@N$ ,  $NDCG@N$  指标上分别最少提升 2.68%, 2.69%, 5.20%. 对于对比模型 AutoCF 而言, LA-MPGCL 在性能和效率方面均较之更优, 在  $Recall@N$ ,  $Precision@N$ ,  $NDCG@N$  指标上分别最少提升 0.31%, 0.81%, 0.78%, 在时间上最少降低 89.58%.

### 3.8 可视化分析 (针对问题 5)

LA-MPGCL 通过有效提取图结构信息和考虑用户与候选项目间的关系, 同时采用层注意力机制聚合最终节点嵌入, 使模型能学习到适度分散的嵌入分布. 为直观展示表征分布的情况, 我们从 LastFM 数据集中随机抽取 1 000 个节点, 并使用 t-SNE 方法将它们的嵌入映射到 2 维空间中, 这些嵌入分布情况如图 8 所示.

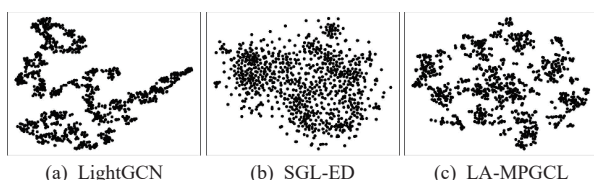


Fig. 8 Visualization of the embedding distribution in LastFM

图 8 LastFM 数据集上嵌入分布的可视化

从图 8 可看出, 经典的非 CL 图方法 (即 LightGCN) 的嵌入分布在嵌入空间中形成难以区分的簇, 这表明其存在过度平滑现象. 相反, 经典的 CL 图方法 (即 SGL-ED) 则呈现大量过于均匀分布的特征, 未

很好地利用全局结构获取用户间的协同关系. 而 LA-MPGCL 在两者之间达到了较好的平衡, 不仅可识别出较清晰的簇形结构, 而且每个簇内的嵌入也合理分散, 能更准确地反映用户的特定偏好. 具体分析如下:

1) 采用层注意力机制代替传统的平均池化, 通过合理分配每一层的聚合权重来减轻高阶图卷积可能对模型带来负面信息的影响, 防止高阶嵌入传播后节点嵌入过于相似的情况.

2) 视图级对比学习中, 通过添加大小可控的噪声来调节表征分布, 并利用 SVD 重构捕获重要的协同信息, 以控制嵌入分布的均匀性.

3) 节点级对比学习中, 利用节点交互的关系构建对比学习, 更好地建模用户的偏好和特征, 学习更高质量的节点嵌入, 以平衡过度平滑与过于均匀现象.

### 3.9 样本案例分析

为直观展示 LA-MPGCL 推荐结果的可解释性, 我们从 LastFM 数据集中选取 5 个用户作为样本, 通过检查用户对项目的交互情况, 同时关注项目标签、社交关系, 观察它们与生成推荐列表的匹配度, 但项目标签与社交关系不参与模型的训练. 我们利用 LA-MPGCL 对其生成 Top-10 项目推荐列表, 并计算其推荐精确率 (实际交互的项目数占推荐列表项目数的比例), 具体如表 7 所示.

**Table 7 Recommendation List Generated by Five Sample Users in LastFM Dataset**

表 7 LastFM 数据集中 5 个样本用户生成的推荐列表

用户	Top-10 推荐列表	精确率
用户 1	[16, <u>237</u> , <u>241</u> , <u>244</u> , 38, 277, 249, 243, 250, 251]	0.4
用户 2	[267, <u>237</u> , 274, <u>238</u> , 206, 473, 411, 240, 248, 636]	0.2
用户 3	[572, 2 218, 3 521, <u>390</u> , 695, 383, <u>3 519</u> , 171, <u>4 276</u> , 1 890]	0.3
用户 4	[ <u>237</u> , 239, 249, <u>241</u> , 243, 244, 298, 416, 267, 411]	0.2
用户 5	[2 829, 530, <u>533</u> , 8 450, 6 647, 7 795, 515, <u>6 737</u> , 532, <u>11 331</u> ]	0.3

注: 加下划线并加粗的序号代表与用户有实际交互的项目.

由表 7 可知, 抽取的 5 个用户节点均给出了较优的预测结果, 且各推荐列表的精确率相差不大, 说明模型稳定性较高、泛化能力较强. 在数据集中, 用户 1 和用户 4 为好友, 2 人喜欢的项目中均含有电子乐、摇滚乐等标签的项目, 为他们推荐项目时, 237, 241 均具有该标签 (即电子乐、摇滚乐). 此外, 为用户 1 和用户 4 推荐的 Top-10 推荐列表中有 5 个项目重合 (即 237, 241, 243, 244, 249), 说明 LA-MPGCL 既在捕获用户共同兴趣方面具有一定优势, 又能根据用户自身的独有特性推荐差异化的项目. 而对非好友用

户推荐的列表差异较大(比如用户 1 和用户 5),说明推荐具有多样性.可见,LA-MPGCL 的推荐性能较好,其推荐结果的可解释性较强.

## 4 总结与下一步工作

本文提出了一种融合层注意力机制的多视角图对比学习推荐方法 LA-MPGCL.该模型通过建立 2 个不同视角(视图级、节点级)下的 3 种对比学习(增强视图对比、候选节点对比、候选结构邻居对比)来获取更高质量的节点嵌入,提升模型的泛化能力.并融合层注意力机制聚合最终节点嵌入,以缓解过度平滑问题.主要工作如下:

1)为提高模型的泛化能力,提出构造多视角图对比学习.通过将加入随机噪声的扰动增强视图与 SVD 的 SVD 增强视图进行视图级对比学习,并通过与基于语义相似度的 2 个节点级对比学习相结合,从而学习更高质量的节点嵌入表示,提升推荐效果.

2)为缓解过度平滑问题,融入层注意力机制聚合.在模型做最终节点嵌入聚合时,采用层注意力机制的方式稳定深层训练,从而使得模型能更高效地利用高阶邻域的图信息改善模型的高阶连通性.

3)根据实验分析所提模型的有效性.在 4 个公开数据集上与相关主流模型进行对比实验,阐明了所提模型的优势.通过消融实验,验证了模型各组件的必要性.开展参数敏感度实验,优化了模型的关键参数.

虽然本文模型在一定程度上优于对比模型,但依旧存在一些不足.所提模型结合视图级和节点级对比学习提升了其泛化能力,利用层注意力机制缓解了过度平滑问题,但未考虑跨层间的对比学习.在下一步工作中,我们将积极探索更多视角的图对比学习方式,以进一步提升模型的泛化能力.

**作者贡献声明:**钱忠胜负责主体思想、整体规划、写作指导;黄恒负责算法设计、数据处理、初稿写作;朱辉负责主要实验及其结果分析;刘金平负责协助实验数据采集、程序设计、文档整理.

## 参 考 文 献

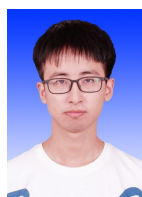
- [1] Chen Huiyuan, Lai V, Jin Hongye, et al. Towards mitigating dimensional collapse of representations in collaborative filtering[C]//Proc of the 17th ACM Int Conf on Web Search and Data Mining. New York: ACM, 2024: 106–115
- [2] Herlocker J L, Konstan J A, Terveen L G, et al. Evaluating collaborative filtering recommender systems[J]. *ACM Transactions on Information Systems*, 2004, 22(1): 5–53
- [3] He Xiangnan, Liao Lizi, Zhang Hanwang, et al. Neural collaborative filtering[C]//Proc of the 26th Int Conf on World Wide Web. New York: ACM, 2017: 173–182
- [4] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback[C]//Proc of the 25th Conf on Uncertainty in Artificial Intelligence. New York: ACM, 2009: 452–461
- [5] 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
- [6] 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
- [7] Qian Zhongsheng, Zhao Chang, Yu Qingyuan, et al. Information fusion recommendation approach combining attention CNN and GNN[J]. *Journal of Software*, 2023, 34(5): 2317–2336 (in Chinese)  
(钱忠胜, 赵畅, 俞情媛, 等. 结合注意力 CNN 与 GNN 的信息融合推荐方法[J]. *软件学报*, 2023, 34(5): 2317–2336)
- [8] Ren Hao, Liu Baisong, Sun Jinyang, et al. A time and relation-aware graph collaborative filtering for cross-domain sequential recommendation[J]. *Journal of Computer Research and Development*, 2023, 60(1): 112–124(in Chinese)  
(任豪, 刘柏嵩, 孙金杨, 等. 基于时间和关系感知的图协同过滤跨域序列推荐[J]. *计算机研究与发展*, 2023, 60(1): 112–124)
- [9] 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
- [10] He Wei, Sun Guohao, Lu Jinhu, et al. Candidate-aware graph contrastive learning for recommendation[C]//Proc of the 46th Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2023: 1670–1679
- [11] Yu Junliang, Yin Hongzhi, Xia Xin, et al. Are graph augmentations necessary? Simple graph contrastive learning for recommendation[C]//Proc of the 45th Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2022: 1294–1303
- [12] Xia Lianghao, Huang Chao, Xu Yong, et al. Hypergraph contrastive collaborative filtering[C]//Proc of the 45th Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2022: 70–79
- [13] Lin Zihan, Tian Changxin, Hou Yupeng, et al. Improving graph collaborative filtering with neighborhood-enriched contrastive learning[C]//Proc of the 31st Int Conf on World Wide Web. New York: ACM, 2022: 2320–2329
- [14] Liu Fan, Cheng Zhiyong, Zhu Lei, et al. Interest-aware message-passing GCN for recommendation[C]//Proc of the 30th Int Conf on World Wide We. New York: ACM, 2021: 1296–1305
- [15] 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

- [16] Wu Zonghan, Pan Shirui, Chen Fengwen, et al. A comprehensive survey on graph neural networks[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2021, 32(1): 4–24
- [17] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[C//OL]//Proc of the 5th Int Conf on Learning Representations. 2017[2024-02-25]. <https://openreview.net/forum?id=SJU4ayYgl>
- [18] Yan Zhao, Xiang Xinguang, Li Zechao. Item correlation modeling in interaction sequence for graph convolutional session recommendation[J]. *SCIENTIA SINICA Informationis*, 2022, 52(6): 1069–1082 (in Chinese)  
(闫昭, 项欣光, 李泽超. 基于交互序列商品相关性建模的图卷积会话推荐[J]. *中国科学: 信息科学*, 2022, 52(6): 1069–1082)
- [19] Li Ting, Jin Fusheng, Li Ronghua, et al. Light-HGNN: Lightweight homogeneous hypergraph neural network for circle content recommendation[J]. *Journal of Computer Research and Development*, 2024, 61(4): 877–888 (in Chinese)  
(李挺, 金福生, 李荣华, 等. Light-HGNN: 用于圈层内容推荐的轻量同质超图神经网络[J]. *计算机研究与发展*, 2024, 61(4): 877–888)
- [20] Berg R, Kipf T N, Welling M. Graph convolutional matrix completion[C]//Proc of the 24th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2018: 974–983
- [21] 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
- [22] Chen Huiyuan, Yeh C M, Wang Fei, et al. Graph neural transport networks with non-local attentions for recommender systems[C]//Proc of the 31st Int Conf on World Wide Web. New York: ACM, 2022: 1955–1964
- [23] Huang Tinglin, Dong Yuxiao, Ding Ming, et al. MixGCF: An improved training method for graph neural network-based recommender systems[C]//Proc of the 27th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2021: 665–674
- [24] Mao Kelong, Zhu Jieming, Xiao Xi, et al. UltraGCN: Ultra simplification of graph convolutional networks for recommendation[C]//Proc of the 30th ACM Int Conf on Information & Knowledge Management. New York: ACM, 2021: 1253–1262
- [25] Shen Yifei, Wu Yongji, Zhang Yao, et al. How powerful is graph convolution for recommendation?[C]//Proc of the 30th ACM Int Conf on Information & Knowledge Management. New York: ACM, 2021: 1619–1629
- [26] Zhang Xiaoyu, Xin Xin, Li Dongdong, et al. Variational reasoning over incomplete knowledge graphs for conversational recommendation[C]//Proc of the 16th ACM Int Conf on Web Search and Data Mining. New York: ACM, 2023: 231–239
- [27] Cao Yang, Gao Min, Yu Junliang, et al. Bi-graph mix-random walk based social recommendation model[J]. *Acta Electronica Sinica*, 2023, 51(2): 286–296 (in Chinese)  
(曹阳, 高旻, 余俊良, 等. 基于双图混合随机游走的社会化推荐模型[J]. *电子学报*, 2023, 51(2): 286–296)
- [28] Yan Mingshi, Cheng Zhiyong, Gao Chen, et al. Cascading residual graph convolutional network for multi-behavior recommendation[J]. *ACM Transactions on Information Systems*, 2023, 42(1): 1–26
- [29] He Kaiming, Fan Haoqi, Wu Yuxin, et al. Momentum contrast for unsupervised visual representation learning[C]//Proc of the 33rd IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 9729–9738
- [30] Cai Xuheng, Huang Chao, Xia Lianghao, et al. LightGCL: Simple yet effective graph contrastive learning for recommendation[C//OL]//Proc of the 11th Int Conf of Learning Representation. 2023[2024-02-25]. <https://openreview.net/forum?id=FKXVK9dyMM>
- [31] Xia Lianghao, Huang Chao, Huang Chunzhen, et al. Automated self-supervised learning for recommendation[C]//Proc of the 32nd Int Conf on World Wide Web. New York: ACM, 2023: 992–1002
- [32] Li Chaoliu, Xia Lianghao, Ren Xubin, et al. Graph transformer for recommendation[C]//Proc of the 46th Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2023: 1680–1689
- [33] Wang Chenyang, Yu Yuanqing, Ma Weizhi, et al. Towards representation alignment and uniformity in collaborative filtering[C]//Proc of the 28th ACM SIGKDD Conf on Knowledge Discovery & Data Mining. New York: ACM, 2022: 1816–1825
- [34] Halko N, Martinsson P G, Tropp J A. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions[J]. *SIAM Review*, 2011, 53(2): 217–288



**Qian Zhongsheng**, born in 1977. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include machine learning, software engineering, and intelligent software.

钱忠胜, 1977年生. 博士, 教授, 博士生导师. CCF高级会员. 主要研究方向为机器学习、软件工程、智能化软件.



**Huang Heng**, born in 1999. Master. His main research interests include intelligent recommender system and software engineering.

黄恒, 1999年生. 硕士. 主要研究方向为智能推荐系统、软件工程.



**Zhu Hui**, born in 1976. PhD candidate. His main research interests include intelligent recommender system and software engineering.

朱辉, 1976年生. 博士研究生. 主要研究方向为智能推荐系统、软件工程.



**Liu Jinping**, born in 1995. PhD candidate. His main research interests include software engineering and intelligent recommender system.

刘金平, 1995年生. 博士研究生. 主要研究方向为软件工程、智能推荐系统.